Universidade do Minho

Database **M**arketing **I**ntelligence **M**ethodology **S**upported by **O**ntologies and **K**nowledge **D**iscovery in **D**atabases

PhD Dissertation

Doctorate in Technology Information and Information systems
Department of Information Systems
University of Minho

PhD Student:

Filipe Mota Pinto

Supervisors:

Prof. Doutor Manuel Filipe Santos

Prof. Doutora Alzira Ascensão Marques

DECLARAÇÃO

Nome

Filipe Jorge Mota Pinto

Endereço electrónico: **fpinto@estg.ipleiria.pt** Telefone: 917766116

Número do Bilhete de Identidade: **928 50 64**

Título da dissertação:

Database Marketing Intelligence Methodology Supported by Ontologies and Knowledge Discovery in Databases

Orientador(es):

Professor Doutor Manuel Filipe Santos

Professora Doutora Alzira Ascensão Marques Ano de conclusão: **2009**

Ramo de Conhecimento do Doutoramento:

Tecnologias e Sistemas de Informação

Universidade do Minho, 27 / 07 / 2009

Assinatura: _____

Dedicated to:

Catarina

Francisco

Linda

Acknowledgments

Thank you very much, Linda! Without you this thesis would never be possible – as simple as this. I love you.

Thank you very much to my wonderful kids, **Catarina** and **Francisco**. With them, sometimes the work would was maybe more complicated to do. However, without them it wouldn´t give any pleasure.

Thank you to my parents, my mother in law and my brother for all shared affection and always available solidarity.

I would like to express my gratitude to my scientific advisors Manuel Filipe Santos and Alzira Marques for giving me the freedom to pursue my interest and explore ideas and to have the "open-mindedness" and confidence in me to see it out.

A very special thanks to my colleague, friend and most of all, partner with whom I've passed through all this time… thank you very much for your support Pedro Gago.

I would also like to express my recognition to the Database Marketing Research Group of Ghent University – Belgium (Professor Dirk Van Der Poel) and to the Protégé Team Research Group of Stanford University – US (Professors Jennifer Vendetti and Samson Tsu) for their support and continuous availability along all this research work.

A special word of recognition to the Polytechnics' president, Prof. Luciano de Almeida, for acknowledging the need to invest in his faculty's advanced research efforts, which falls in line with the new strategic plans for our sustainability as an Institute. Also here, I would like to express my gratitude to Paulo Bartolo who had coordinated the UM-IPLeiria PhD program and also motivated each one in order to achieve success.
The financial support given from Fundação para a Ciência e Tecnologia (FCT), through the PhD scholarship with reference SFRH/BD/36541/2007, is gratefully acknowledged.

Resumo

Actualmente as organizações actuam em ambientes caracterizados pela inconstância, elevada competitividade e pressão no desenvolvimento de novas abordagens ao mercado e aos clientes. Nesse contexto, o acesso à informação, o suporte à tomada de decisão e a partilha de conhecimento tornam-se essenciais para o desempenho organizativo.

No domínio do marketing têm surgido diversas abordagens para a exploração do conteúdo das suas bases de dados. Uma das abordagens, utilizadas com maior sucesso, tem sido o processo para a descoberta de conhecimento em bases de dados. Por outro lado, a necessidade de representação e partilha de conhecimento tem contribuído para um crescente desenvolvimento das ontologias em áreas diversas como sejam medicina, aviação ou segurança.

O presente trabalho cruza diversas áreas: tecnologias e sistemas de informação (em particular a descoberta de conhecimento), o marketing (especificamente o database marketing) e as ontologias. O objectivo principal desta investigação foca o papel das ontologias em termos de suporte e assistência ao processo de descoberta de conhecimento em bases de dados num contexto de database marketing. Através de abordagens distintas foram formuladas duas ontologias: ontologia para o processo de descoberta de conhecimento em bases de dados e, a ontologia para o processo database marketing suportado na extracção de conhecimento em bases de dados (com reutilização da ontologia anterior). O processo para licitação e validação de conhecimento, baseou-se no método de Delphi (ontologia de database marketing) e no processo de investigação baseada na revisão de literatura (ontologia de descoberta de conhecimento). A concretização das ontologias suportou-se em duas metodologias: metodologia methontology, para a ontologia de descoberta de conhecimento e metodologia 101 para a ontologia de database marketing. A última, evidencia a reutilização de ontologias, viabilizando assim a reutilização da ontologia de descoberta de conhecimento na ontologia de database marketing. Ambas ontologias foram

desenvolvidas sobre a ferramenta Protege-OWL permitindo não só a criação de toda a hierarquia de classes, propriedades e relações, como também, a realização de métodos de inferência através de linguagens baseadas em regras de Web semântica. Posteriormente, procedeu-se à experimentação da ontologia em casos práticos de extracção de conhecimento a partir de bases de dados de marketing.

O emprego das ontologias neste contexto de investigação, representa uma abordagem pioneira e inovadora, uma vez que são propostas para assistirem em cada uma das fases do processo de extracção de conhecimento em bases de dados através de métodos de inferência. È assim possível assistir o utilizador em cada fase do processo de database marketing em acções tais como de selecção de actividades de marketing em função dos objectivos de marketing (e.g., perfil de cliente), em acções de selecção dados (e.g., tipos de dados a utilizar em função da actividade a desenvolver) ou mesmo no processo de selecção de algoritmos (e.g. inferir sobre o tipo de algoritmo a usar em função do objectivo definido).

A integração das duas ontologias num contexto mais lato permite, propor uma metodologia com vista ao efectivo suporte do processo de database marketing baseado no processo de descoberta de conhecimento em bases de dados, denominado nesta dissertação como: Database Marketing Intelligence. Para a demonstração da viabilidade da metodologia proposta foi seguido o método *action-research* com o qual se observou e testou o papel das ontologias no suporte à descoberta de conhecimento em bases de dados (através de um caso prático) num contexto de database marketing. O trabalho de aplicação prática decorreu sobre uma base de dados real relativa a um cartão de fidelização de uma companhia petrolífera a operar em Portugal.

Os resultados obtidos serviram para demonstrar em duas vertente o sucesso da abordagem proposta: por um lado foi possível formalizar e acompanhar todo o processo de descoberta de conhecimento em bases de dados; por outro lado, foi possível perspectivar uma metodologia para um domínio concreto suportado por ontologias (suporte á decisão na selecção de métodos e tarefas) e na descoberta de conhecimento em bases de dados.

Abstract

Nowadays, the environment in which companies work is turbulent, very competitive and pressure in the development of new approaches to the market and clients. In this context, the access to information, the decision support and knowledge sharing become essential for the organization performance.

In the marketing domain several approaches for the exploration of database exploration have emerged. One of the most successfully used approaches has been the knowledge discovery process in databases. On the other hand, the necessity of knowledge representation and sharing and contributed to a growing development of ontologies in several areas such as in the medical, the aviation or safety areas.

This work crosses several areas: technology and information systems (specifically knowledge discovery in databases), marketing (specifically database marketing) and ontologies in general. The main goal of this investigation is to focus on the role of ontologies in terms of support and aid to the knowledge discovery process in databases in a database marketing context. Through distinct approaches two ontologies were created: ontology for the knowledge discovery process in databases, and the ontology for the database marketing process supported on the knowledge extraction in databases (reusing the former ontology). The elicitation and validation of knowledge process was based on the Delphi method (database marketing ontology) and the investigation process was based on literature review (knowledge discovery ontology). The carrying out of both ontologies was based on two methodologies: methontology methodology, for the knowledge discovery process and 101 methodology for the database marketing ontology. The former methodology, stresses the reusing of ontologies, allowing the reusing of the knowledge discovery ontology in the database marketing ontology. Both ontologies were developed with the Protege-OWL tool. This tool allows not only the creation of all the hierarchic classes, properties and relationships, but also the carrying out of inference methods through web semantics based languages. Then, the ontology was tested in practical cases of knowledge extraction from marketing databases.

The application of ontologies in this investigation represents a pioneer and innovative approach, once they are proposed to aid and execute an effective support in each phase of the knowledge extraction from databases in the database marketing context process. Through inference processes on the knowledge base created it was possible to assist the user in each phase of the database marketing process such as, in marketing activity selection actions according to the marketing objectives (e.g., client profile) or in data selection actions (e.g., type of data to use according to the activity to be preformed. In relation to aid in the knowledge discovery process in databases, it was also possible to infer on the type of algorithm to use according to the defined objective or even according to the type of data pre-processing activities to develop regarding the type of data and type of attribute information.

The integration of both ontologies in a more general context allows proposing a methodology aiming to the effective support of the database marketing process based on the knowledge discovery process in databases, named in this dissertation as: Database Marketing Intelligence. To demonstrate the viability of the proposed methodology the action-research method was followed with which the role of ontologies in assisting knowledge discovery in databases (through a practical case) in the database marketing context was observed and tested. For the practical application work a real database about a customer loyalty card from a Portuguese oil company was used.

The results achieved demonstrated the success of the proposed approach in two ways: on one hand, it was possible to formalize and follow the whole knowledge discovery in databases process; on the other hand, it was possible to perceive a methodology for a concrete domain supported by ontologies (support of the decision in the selection of methods and tasks) and in the knowledge discovery in databases.

Table of Contents

List of Figures

List of Tables

Glossary

API .....Application Program Interface

DARPA .....Defense Advanced Research Projects Agency

DAML DARPA .....Agent Markup Language

DBM .....Database Marketing

DBMI .....Database Marketing Intelligence

DBMO.....Database Marketing Ontology

DL .....Description Logic

FLogic .....Frame Logic

FTP .....File Transfer Protocol

HTML .....HyperText Markup Language

HTTP .....HyperText Transfer Protocol

NS .....NameSpace

OIL .....Ontology Inference Layer

OWL .....Web Ontology Language

KDD .....Knowledge Discovery in Databases

RDF .....Resource Description Framework

RDFS .....Resource Description Framework Schema

SHOE .....Simple HTML Ontology Extensions

SPARQL .....Simple Protocol and RDF Query Language

SWRL .....Semantic Web Rule Language

URI .....Uniform Resource Identifier

URL .....Uniform Resource Locator

W3C .....World Wide Web Consortium

WWW.....Web World Wide Web

XML .....eXtensible Markup Language

XMLS .....eXtensible Markup Language Schema

I

Introduction

Database Marketing (DBM), as a discipline, operates using a range of different approaches and methods in order to use and explore, marketing databases as much as possible. Those approaches and methods, throughout statistical and simulation modules include, amongst others, database management skills, data analysis expertise or knowledge judgment capacity. In this context, Knowledge Discovery in Databases (KDD) is introduced. KDD uses a range of methods and tasks that aim to get new and useful knowledge from databases, here related as marketing databases. In spite of such definitions, there is a gap of communication and knowledge share between DBM practitioners and KDD analysts. Whenever one needs the other, there are a set of redundant procedures that could be optimized through knowledge sharing. To overcome this, ontologies may play an important role through DBM knowledge base creation. Ontologies aim to capture consensual knowledge in a generic way, and that they may be reused and shared across either software applications and by groups of people.

This work bridges three disciplines: information systems and technology (more specifically, knowledge extraction from databases), ontologies and marketing (more specifically, database marketing). An interdisciplinary research was made based on ontology techniques (planning, conceptualization, axiomatization and knowledge base creation), applied to DBM. Ontologies were used to structure and modulate DBM related knowledge and therefore to identify pointers for knowledge extraction techniques. Indeed, we use ontologies to assist the interactive KDD process during its development. Closing the research, we have created a system design prototype, the DBMi (Database Marketing Intelligence), that will use the Database Marketing

Ontology (DBMO) to guide the DBM process and to assist the KDD process throughout KDD ontology.

The ontology assistance to the KDD process development provides a more flexible management capability to the marketers and data analysts. Nevertheless, the methodology is also of general interest, given that its "ontology-based" architecture can be applied to any DBM project supported by the KDD process, at an appropriate level of abstraction. Hence, DBMO provides marketers with useful insights for DBM process development and support, assisting them in the process guidance and task selection at each KDD phase.

The DBMi improves both speed and efficiency for the DBM process, through the knowledge reuse in the KDD process assistance. Also, it provides support to marketing in complex problem-solving, facilitates knowledge modeling and reuse by means of DBMO.

The following research targets have been achieved in particular: (i) taxonomy construction of knowledge extraction process from databases – KDD ontology; (ii) modeling information about marketing databases exploration processes – Database Marketing Ontology; (iii) development of a system prototype to the effective KDD ontological assistance; (iv) development of a database marketing supported by KDD methodology - Database Marketing Intelligence.

1.1 Motivation

Knowledge discovery in databases is a well accepted definition for related methods, tasks and approaches for knowledge extraction activities (Brezany *et al.*, 2008) (Nigro *et al.*, 2008). Knowledge extraction is also referred as a set of procedures that cover all work ranging from data collection to algorithms execution and model evaluation. In each of the development phases, practitioners employ specific methods and tools that support them in fulfilling their tasks. The development of methods and tasks for the different disciplines have been established and used for a long time (Domingos, 2003) (Cimiano *et al.*, 2004) (Michalewicz *et al.*, 2006). Until recently, there was no need to integrate them in a structured manner (Tudorache, 2006). However, with the wide use of this approach, engineers were faced with a new challenge: They had to deal with a multitude of heterogeneous problems originating from different approaches and had to make sure that in the end all models offered a coherent business domain output. There are no mature processes and tools that enable the exchange of models between the different parallel developments at different contexts (Jarrar, 2005). Indeed, there is a gap in the KDD process knowledge sharing in order to promote its reuse.

The Internet and open connectivity environments created a strong demand for the sharing of data semantics (Jarrar, 2005). Emerging ontologies are increasingly becoming essential for computer science applications. Organizations are beginning to view them as useful machine-processable semantics for many application areas. Hence, ontologies have been developed in artificial intelligence to facilitate knowledge sharing and reuse. They are a popular research topic in various communities, such as knowledge engineering (Borst *et al.*, 1997) (Bellandi *et al.*, 2006), cooperative information systems (Diamantini *et al.*, 2006b), information integration (Bolloju *et al.*, 2002) (Perez-Rey *et al.*, 2006), software agents (Bombardier *et al.*, 2007), and knowledge management (Bernstein *et al.*, 2005) (Cardoso and Lytras, 2009). In general, ontologies provide (Fensel *et al.*, 2000): a shared and common understanding of a domain which can be communicated amongst people and across application systems; and, an explicit conceptualization (i.e., meta information) that describes the semantics of the data.

Nevertheless, ontological development is mainly dedicated to a community (e.g., genetics, cancer or networks) and, therefore, is almost unavailable to others outside it. Indeed the new knowledge produced from reused and shared ontologies is still very limited (Guarino, 1998) (Blanco *et al.*, 2008) (Coulet *et al.*, 2008) (Sharma and Osei-Bryson, 2008) (Cardoso and Lytras, 2009).

To the best of our knowledge, in spite of successful ontology approaches to solve some KDD related problems, such as, algorithms optimization (Kopanas *et al.*, 2002)(Nogueira *et al.*, 2007), data pre-processing tasks definition (Bouquet *et al.*, 2002)(Zairate *et al.*, 2006) or data mining evaluation models (Cannataro and Comito, 2003)(Brezany *et al.*, 2008), the research to the ontological KDD process assistance is sparse and spare. Moreover, mostly of the ontology development focusing the KDD area focuses only a part of the problem, intending only to modulate data tasks (Borges *et al.*, 2009), algorithms (Nigro *et al.*, 2008), or evaluation models (Euler and Scholz, 2004)(Domingues and Rezende, 2005). Also, the use of KDD in marketing field has been largely ignored (with a few exceptions (Zhou *et al.*, 2006)(El-Ansary, 2006)(Cellini *et al.*, 2007)). Indeed, many of these works provide only single specific ontologies that quickly become unmanageable and therefore without the sharable and reusable characteristic. Such research direction may became innocuous, requiring tremendous patience and an expert understanding of the ontology domain, terminology, and semantics.

Contrary to this existing research trend, we feel that since the knowledge extraction techniques are critical to the success of database use procedures it follows in which researchers are interested in addressing the problem of knowledge share and reuse. We must address and emphasize the knowledge conceptualization and specification through ontologies.

Therefore, this research promises interesting results in different levels, such as:
Regarding information systems and technologies, focusing the introduction and integration of the ontology to assist and improve the KDD process, through inference tasks in each phase;

In the ontology area this investigation represents an initial approach step on the way for real portability and knowledge sharing of the system towards other similar DBM process supported by the KDD process. It could effectively be employed to address the general problem of model-construction in problems similar to the one of marketing (generalization), on the other side it is possible to instantiate/adapt the ontology to the specific configuration of a DBM case and to automatically assist, suggest and validate specific approaches or models KDD process (specification);

Lastly, for data analyst practitioners this research may improve their ability to develop the DBM process, supported by KDD. Since knowledge extraction work depended in large scale on the user background, the proposed methodology may be very useful when dealing with complex marketing database problems. Therefore the introduction of an ontological layer in DBM project allows: more efficient and stable marketing database exploration process through an ontology-guided knowledge extraction process; and, portability and knowledge share among DBM practitioners and computer science researchers.

1.2 Research Objectives and Contribution

The main general purpose of this research is to explore ontologies in order to improve KDD approach applied to DBM problems. In this project, the study of different approaches for DBM through knowledge extraction methods and techniques is aimed at enhancing the competence of ontologies as a guide. Therefore, the main question investigated by this thesis is:

How ontologies may facilitate the process of knowledge discovery from marketing databases through the database marketing process?

Besides the main research question and effective contribution, this work also contributes with:

- A DBM ontology proposal;
- A KDD ontology proposal;
- A system prototype for ontologies integration and assistance to the KDD process within the DBM projects.

In order to achieve this, the following tasks have been accomplished:

- *Knowledge Discovery Process Ontology*: KDD process is already accepted by a vast computer researcher's community. Here we use its general framework to modulate knowledge of all related processes and techniques;
- *Database marketing process systematization*: since the marketing discipline is, more than ever, technology dependent, the marketing database usage is available for a widespread of approaches and techniques. Therefore we have carried out some knowledge elicitation tasks in order to achieve, among experts, a consensual database marketing framework;
- *Database marketing ontology knowledge structure*: since database marketing is the application domain of this research, we have formulated an ontology in order to create a related knowledge base. Moreover, we have reused the previously designed KDD ontology following the 101 methodology;

- *Experimental KDD complete running process*: in order to test and experiment the designed KDD ontology we have made some knowledge discovery work over a real marketing database;
- *Methodology prototype design*: we have integrated previous designed ontologies in a multi layer system in order to formulate a general framework in the context of database marketing. Moreover, we have considered inter-layer activities supporting inference actions and user interface;

This dissertation does not attempt to create an effective and programmed system based on ontologies and knowledge discovery in databases. Our intention is to offer a methodology framework in which such theories can be used to help the integration both types of information: ontological and knowledge discovery in databases.

1.3 Structure of the Thesis

We start with a generic focus (Figure 1), presenting the motivation and related knowledge. Then we focus our work through the research contribution. In the closing part of this work discussion and conclusions are presented, tending to be not only rigorous but also generic enough to address future researches.

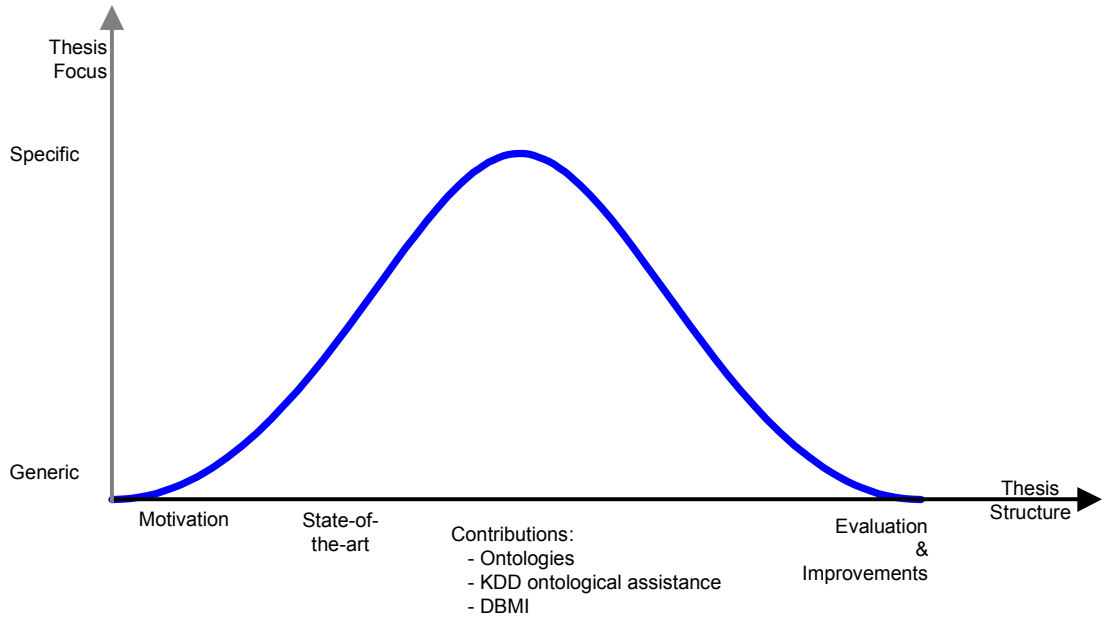


Figure 1: Dissertation focus

This document is organized in four main parts, containing ten chapters (Figure 2).

The *first part* (the introductory part), is composed of the first chapter which is dedicated to fundamental motivation for this work, main objectives presentation and dissertation structure explanation.

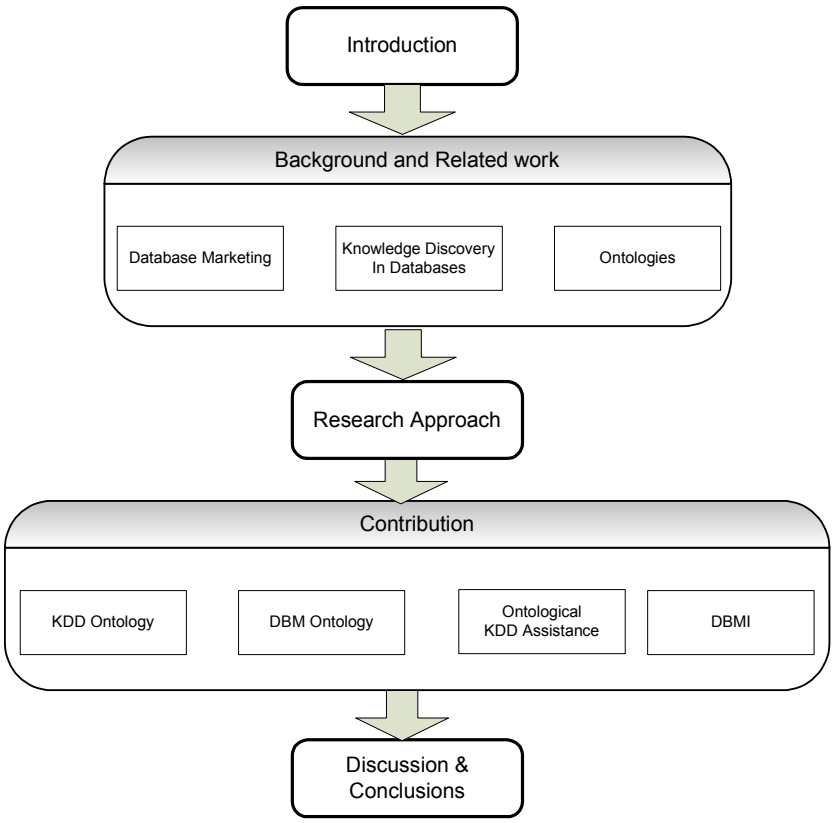


Figure 2: Dissertation structure

The *second part* is composed of three chapters which cover *background knowledge* and *related work*. The first covers all related marketing issues concerning DBM; then a literature review focusing knowledge extraction over databases is presented, next, in the third chapter an ontology literature review. In the closing section of this part, the research approach taken in this work is introduced and justified.

The *third part* of this document is dedicated to the *developed work* and *contribution*, whereas *KDD ontology, DBM ontology*, *Ontological KDD assistance* and *DBMI system prototype* are presented. This part concerns the ontology development, conception, development (knowledge base) and deployment (system prototype) - a vision of the ontology body as an integrated framework which acts with a knowledge base in order to guide and support the knowledge extraction process in DBM projects.

The remaining part of the dissertation concludes the thesis where the research questions, raised in the initial section, are answered and tasks are solved, and includes the answer to the central question "*How ontologies may facilitate the process of knowledge discovery at marketing databases through the database marketing process?"*. Also current and future research and engineering directions are summarized.

II

Background and Related Work

This chapter gives an introduction to the subject matter and to the background of this research work. First of all, the KDD is presented through the process framework. Several methods and techniques are briefly outlined, and Data mining (DM) is introduced as the core step within the KDD process. Following this, we focus the DBM regarding basic concepts and definitions. We also introduce the DBM process supported by KDD process. Then, ontologies state-of-the-art is reviewed, with focus on the methodologies, languages and tools. Ending this chapter, related work is presented.

2.1 Knowledge Discovery in Databases

KDD is commonly defined as "the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Fayyad *et al.*, 1996).

The term "data" is understood as a set of facts or atomic pieces of information (e.g. cases in a database) while "knowledge" stands for a higher-level concept that relates to the properties of the collection of data as a whole (e.g. dependencies among sets of attributes in a database and rules for predicting attribute values).

KDD has succeeded very well in marketing field with special focus on direct marketing activities (Buckinx and den Poel, 2005, Buckinx *et al.*, 2007, den Poel and Buckinx, 2005). Indeed, this marketing research area has become an important application field for DM (.e.g., companies or organizations try to establish and maintain a direct relationship with their customers in order to target them individually for specific product offers or fund raising, throughout marketing databases exploration).

2.1.1 Goals and Themes of KDD

According to the literature, KDD has two main goals which are oriented by users' intensions (Han and Kamber, 2001) (Kuo *et al.*, 2007a):

- *Prediction*: using available data to predict unknown or future values giving some variables. The main goal of the prescriptive process effort is to automate a decision making process by creating a model capable of making a prediction, assigning a label, or estimating a value. Normally, the model results will be acted upon directly, which makes accuracy the most important measure of performance when evaluating this type of models;
- *Description*: The primary goal of descriptive data mining is to gain increased understanding of the data in order to find some interesting patterns and presenting it to the user in an easily understood way. Although it often results in actions, these are not the sort of actions that can be automated directly from the results of the model. Besides that, the best model may not be the one that makes the most accurate predictions. Often the insight gained through building the model is the most important part of the process, and the actual results from the model may never be used at all.

As main distinction between prediction and description is who interprets the discovered knowledge – the system (in case of prediction) or the user (in case of description) (Fayyad *et al.*, 1996, Piatetsky-Shapiro, 2007, Piatetsky-Shapiro, 1991). However, the

boundary between these two goals is not distinct since some predictive models can be used for description and vice versa (Piatetsky-Shapiro, 1991).

KDD has connections with many research fields, such as statistics, database theory and artificial intelligence techniques. Therefore, research themes in KDD are scattered across a range of topics including (Ankerst *et al.*, 2003)(Piatetsky-Shapiro, 2007): data representation, large databases, model pruning and simplification, visualization and quality assessment.

Other areas in KDD may include decomposition of the process, development of discretization methods, other pre-processing techniques (in order to ensure data quality) or development of parallel steps in the KDD process, among others.

2.1.2 Knowledge Extraction Integrated Process

The overall KDD process is depicted in Figure 3. It consists of several steps and phases that are interactive and iterative with many decisions being made by the users (Fayyad and Uthurusamy, 1996). From a data source containing raw data, all or portions of the data are selected for further processing. The selected raw data – target data - is then typically pre-processed and transformed in some way, before being passed on to the data mining algorithm itself. The patterns output from the mining procedure are then post-processed, interpreted and evaluated, hopefully revealing new knowledge contained in the data. Along the way, in practice backtracking on each step will inevitably occur.

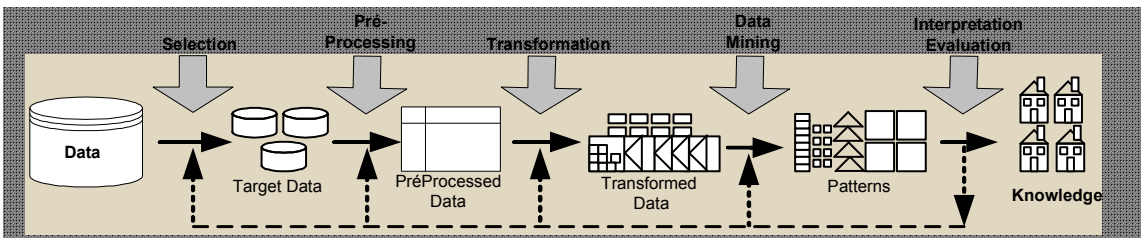


Figure 3: Knowledge discovery process, adopted from (Fayyad *et al.*, 1996)

The main objective of KDD is to transform data into actionable results. The basic steps to do this are: Identifying the problem; Transforming data into actionable results; Acting on results; Measuring the results.

The step, *identifying the problem*, is a difficult part since it is exclusively a communication problem from data analysts and domain people. Data analyst need to understand the details of the problem and the goals of the data mining process, since even the most advanced algorithms cannot figure out what is most important.
However the heart of KDD is to transform data into actionable results. The basic steps to do this are (Figure 3):

- Data selection. The first step in the modeling process is identifying and obtaining the relevant data. Often, the relevant data is simply whatever data is available, reasonably clean, and accessible. It is important to verify that the data meets the requirements for solving the problem, and is as complete as possible before starting the modeling, e.g., when the purpose of the data mining effort is to identify customer segments, for the purpose of directly addressing a purchasing list to prospective customers. In this case, the data needs to contain fields that are appropriate for purchasing advertising space and lists, such as location and demographic information. The data also needs to contain the desired outcome, and it is important to keep in mind that, knowing who has responded to the previous marketing campaigns, without knowing who had been contacted, is almost useless. In fact, without knowing who had been contacted it is impossible to know if a certain customer is a non-responder because he has not been targeted by the direct marketing campaign, or because he really was not interested in the product and did not respond to the product offer. Note that only the customers who had been contacted in the previous marketing campaigns should be used as examples to build the predictive models, since the other will all be considered non-responders even though they could have responded positively if they had been contacted;

- *Data pre-processing,* intends to validate, explore and clean the data. Such tasks will find the answers to the following questions (Madeira, 2002): Are the fields

populated?; Will missing values be a problem? Are the field values legal?. That is: Are numeric fields within proper bounds and are code fields all valid? Are the field values reasonable? Is the distribution of individual fields explainable? Always keep in mind that the outcome of data mining depends critically on the data, and that data inaccuracies creep in from many different places;

- *Data transformation*, is developed in order to transpose the data to the right granularity. Granularity refers to the level of the data being modeled. Nowadays, all the data mining algorithms work on individual rows of data. So, all the data describing a customer (or whatever one is interested in) must be in a single row. All the data available must be summarized to the right level of granularity – a single table where each row represents a data object. Also new attributes derivation is included at this step. Derived variables are variables which values are based on combinations of other values inside the data;

- *Data mining*, begins with data set preparation that will be used to actually build the data mining models. Once the data has been cleaned, transposed, and the necessary derived variables added, there are a few things that we still have to take into account, e.g., when we are building a predictive model from historical data, then we should find out which should be the frequency of the rarer outcomes in the model set. A good rule of thumb seems to be that, the data should have between 15% and 30% density of the rarer outcomes. As this point, we also need to decide the way we are going to evaluate the performance of the model. Only some of the data should be used to create the model and the other data should be held in batch to refine the model, and to predict how well it works on unseen data. On one hand, we can decide to divide the data into three sets: training, testing and evaluation sets. In this case, the first two are used to train and refine the model, while the third is used to evaluate the model on unseen examples. On the other hand, one can opt to divide the model set into two sets only. The first is then used to train using n-fold cross-validation. The second one is used as an independent test set, which is then used to assess the performance of the model on unseen examples. Note that in this case, the model is refined during the cross-validation process.

Within data mining we have to choose the modeling technique and train the model. There are a variety of different data mining techniques, such as statistical regression, decision trees, neural network, or fuzzy modeling to choose from. Each one has its advantages and disadvantages, so the best one depends on the problem. When time is available, different techniques can be applied to the same model set in order to choose the one that produces the best models. Note that the parameterization of training a model depend on the chosen data mining technique, on the learning algorithm which implements it, and on the tool being used;

- *Interpretation and evaluation*. The final step is to validate the different models on the data. This is done by checking performance of each model on data that was not used to derive it. At this stage, one should have in mind that, the best model depends always on the specific problem. Although different data mining techniques have different ways of measuring the results, we always want to compare their performance on unseen data. To do so, the assessment must be done using the data that was not used to train the model, and the methods described further can be used. A confusion matrix, which tells us how many predictions made by a predictive model are correct/incorrect, and how did the model classify the examples belonging to the different classes, can also be used.

However, data mining serves no purpose if we never act on the results of the model (Gersten *et al.*, 2000). Acting on the results can take several different forms (Giudici and Passerone, 2002). On one hand, during the modeling, we may have learned new facts from data, which may lead to insights about the specifics of the business, the customers, or the structure of the system modeled. On the other hand, the results may be focused on a particular activity, such as a marketing campaign, which should then be carried out, based on the propensities determined by the model (One-Time results). Nevertheless, in the specific case of target selection models with marketing purposes, the model results provide interesting information about customers (NG and LIU, 2000) (Rygielski *et al.*, 2002) that should then be accessible to the company (remembered results), and may also be used to score customers periodically, to determine who should be the targets of the next marketing campaign, or which is the best offer to make them next (periodic

predictions). The model itself may be incorporated into another system to provide real-time predictions (Real-time scoring). Lastly, the model results can be used to fix the data. In fact, sometimes, the data mining effort uncovers data problems that significantly affect the performance of the models (Naik and Tsai, 2004).

2.1.3 Data Mining Functions

The general data mining tools, classified by data mining tasks, are presented below (Sarker *et al.*, 2002):

Table 1: Data mining tools by tasks

Mining Task	Data Mining Tool
Feature Selection	Dependency Models
	Optimization
Prediction/Estimation	Regression models
	Neural networks
	Regression decision
	Support vector machines
	Optimization
Classification	Statistical regression models
	Neural networks
	Decision trees
	Support vector machines
	Optimization
Rule Discovery	Decision trees
	Optimization
	Learning classifiers systems
Clustering	Density estimation methods
	Neural networks
	Clustering
	Optimization
Association	Association rules
	Density estimation models
	Optimization

As presented in Table 1, the basic functions of the data mining process include feature selection, summarization, association, clustering, prediction, and classification. These are summarized below:

Feature Selection concerns the identification of a subset of features that significantly contributes to the discrimination or prediction problem (Prinzie and Poel, 2006). Feature selection problem is also presented as a mathematical program with a parametric objective function and linear constraints (NG and LIU, 2000). Another approach uses a very fast iterative linear-programming-based algorithm for solving the problem that terminates in a finite number of steps (Kim, 2008).

Summarization involves methods for finding a compact description for a subset of data. Summarization can be performed using a bar chart or statistical analysis. This is useful for understanding the importance of certain attributes when compared with each other (Witten and Frank, 2000). More sophisticated methods involve the derivation of summary rules (Agrawal *et al.*, 1993), multivariate visualization techniques, and the discovery of functional relationships between variables (Changchien and C., 2001).

Association rules determine how the various attributes are related. The association rules are also known as Dependency Modeling, which exist in two levels: the structural level of the model specifies (often in graphical form) which variables are locally dependent on which, whereas the quantitative level of the model specifies the strengths of the dependencies using some numerical scale (Fayyad *et al.*, 1996) (Piatetsky-Shapiro, 2007).

Clustering identifies a finite set of categories or clusters to describe the data (Breiman *et al.*, 1984) (Santos *et al.*, 2005). The categories may be mutually exclusive and exhaustive, or consist of a richer representation such as hierarchical or overlapping categories (Fayyad and Uthurusamy, 1996). Unlike classification, the number of desired groups is unknown. As a result, the clustering problem is usually treated as a two-stage optimization problem. In the first stage, the number of clusters is determined followed by the next stage, where the data is assigned to the best possible cluster. However, one needs to be careful here, as this type of sequential optimization techniques does not guarantee the optimality of the overall problem. The use of regression modeling for

point estimation is basically an unconstrained optimization problem that minimizes an error function. Artificial Neural Networks are widely used for prediction, estimation and classification (Witten and Frank, 2000) (Silva *et al.*, 2004). In terms of model evaluation, the standard squared error and cross entropy loss functions for training artificial neural networks can be viewed as log-likelihood functions for regression and classification respectively (Kuo *et al.*, 2007a) (Nigro *et al.*, 2008). Regression Trees and Rules are also used for predictive modeling, although they can be applied for descriptive modeling as well (Lariviere and den Poel, 2005) (Vindevogel *et al.*, 2005) .

In *classification*, the basic goal is to predict the most likely state of a categorical variable (the class) given the values of the other variables. This is fundamentally a density estimation problem (Kurt *et al.*, 2008). A number of studies have been undertaken in the literature for modeling classification as an optimization problem (Kamber *et al.*, 1997) including discriminant analysis for classification (Berson and Smith, 2001) which uses an unconstrained optimization technique for error minimization (Anand *et al.*, 2007) (Lin and Hong, 2008).

Rule discovery is one of the most important data mining tasks (Yohannes and Hoddinott, 1999). The basic idea is to generate a set of symbolic rules that describe each class or category. Rules should usually be simple to understand and interpret. Rule discovery can be a natural outcome of the classification process as a path in a decision tree from the root node to a leaf node represents a rule (Sarker *et al.*, 2002). However, redundancy is often present in decision trees (Quinlan, 1986) and the extracted rules are always simpler than the tree (Kurt *et al.*, 2008). It is also possible to generate the rules directly without building a decision tree as an intermediate step. In this case, learning classifier systems play a key method is rule discovery (Shi *et al.*, 2006).

2.2 Database Marketing

Much of the advanced practice in Database Marketing (DBM) is performed within private organizations (Zwick and Dholakia, 2004)(Marsh, 2005). This may partly explain the lack of articles published in the academic literature that study DBM issue (Bohling *et al.*, 2006)(Frankland, 2007)(Lin and Hong, 2008).

However, DBM is nowadays an essential part of marketing in many organizations. Indeed, as the main DBM principle, most organizations should communicate as much as possible with their customers on a direct basis (DeTienne and Thompson, 1996). Such objective has contributed to the expressive grown of all DBM discipline. In spite of such evolution and development, DBM has growth without the expected maturity (Fletcher *et al.*, 1996) (Verhoef and Hoekstra, 1999).

In some organizations, DBM systems work only as a system for inserting and updating data, just like a production system (Sen and Tuzhiln, 1998). In others, they are used only as a tool for data analysis (Bean, 1999). In addition, there are corporations that use DBM systems for both operational and analytical purposes (Arndt and Gersten, 2001). Currently DBM is mainly approached by classical statistical inference, which may fail when complex, multi-dimensional, and incomplete data is available (Santos *et al.*, 2005).

One of most cited origins of DBM is the retailers' catalogue based in the USA selling directly to customers. The main means used was direct mail, and mailing of new catalogues usually took place to the whole database of customers (DeTienne and Thompson, 1996). Mailings result analysis has led to the adoption of techniques to improve targeting, such as CHAID (Chi-Squared Automated Interaction Detection) and logistic regression (DeTienne and Thompson, 1996) (Schoenbachler *et al.*, 1997). Lately, the addition of centralized call centers and the Internet to the DBM mix has introduced the elements of interactivity and personalization. Thereafter, during the 1990s, the data-mining boom popularized such techniques as artificial neural networks, market basket analysis, Bayesian networks and decision trees (Pearce *et al.*, 2002) (Drozdenko and Perry, 2002).

2.2.1 Definition

DBM refers to the use of database technology for supporting marketing activities(Leary *et al.*, 2004)(Wehmeyer, 2005)(Pinto *et al.*, 2009). Therefore, it is a marketing process driven by information (Coviello *et al.*, 2001)(Brookes *et al.*, 2004) (Coviello *et al.*, 2006) and managed by database technology (Carson *et al.*, 2004) (Drozdenko and Perry, 2002). It allows marketing professionals to develop and to implement better marketing programs and strategies (Shepard, 1998) (Ozimek, 2004).

There are different definitions of DBM with distinct perspectives or approaches denoting some evolution an evolution along the concepts (Zwick and Dholakia, 2004). From the marketing perspective, DBM is an interactive approach to marketing communication. It uses addressable communications media (Drozdenko and Perry, 2002) (Shepard, 1998), or a strategy that is based on the premise that not all customers or prospects are alike. By gathering, maintaining and analyzing detailed information about customers or prospects, marketers can modify their marketing strategies accordingly (Tao and Yeh, 2003). Then, some statistical approaches were introduced and DBM was presented as the application of statistical analysis and modeling techniques to computerized individual level data sets (Sen and Tuzhiln, 1998)(Rebelo *et al.*, 2006) focusing some type of data. Here, DBM simply involves the collection of information about past, current and potential customers to build a database to improve the marketing effort (Brito and Hammond, 2007). The information includes: demographic profiles; consumer likes and dislikes; taste; purchase behavior and lifestyle (Seller and Gray, 1999) (Pearce *et al.*, 2002).

As information technologies improved their capabilities such as processing speed, archiving space or, data flow in organizations that have grown exponentially different approaches to DBM have been suggested: generally, it is the art of using data you've already gathered to generate new money-making ideas (Gronroos, 1994) (Pearce *et al.*, 2002); stores this response and adds other customer information (lifestyles, transaction history, etc.) on an electronic database memory and uses it as basis for longer term

customer loyalty programs, to facilitate future contacts, and to enable planning of all marketing. (Fletcher *et al.*, 1996)(Kurtulus and Kurtulus, 2006)(Frankland, 2007); or, DBM can be defined as gathering, saving and using the maximum amount of useful knowledge about your customers and prospects...to their benefit and organizations' profit. (McClymont and Jocumsen, 2003) (Pearce *et al.*, 2002). Lately some authors has referred DBM as a tool database-driven marketing tool which is increasingly taking centre stage in organizations strategies (Payne and Frow, 2005)(Pinto, 2006)(Lin and Hong, 2008).

In common all definition share a main idea: DBM is a process that uses data stored in marketing databases in order to extract relevant information to support marketing decision and activities through customer knowledge, which will allow satisfy their needs and anticipate their desires.

2.2.2 Development

During the DBM process it is possible to consider three phases (DeTienne and Thompson, 1996)(Shepard, 1998)(Drozdenko and Perry, 2002): data collection, data processing (modeling) and results evaluation.

Figure 4 presents a simple model of how customer data are collected through internal or external structures that are closer to customers and the market, how customer data is transformed into information and how customer information is used to shape marketing strategies and decisions that later turn into marketing activities. The first, *Marketing data*, consists in data collection phase, which will conduct to marketing database creation with as much customer information as possible (e.g., behavioral, psychographic or demographic information) and related market data (e.g., share of market or competitors information's). During the next phase, *information*, the marketing database is analyzed under a marketing information perspective throughout activities such as, information organization (e.g., according organization structure, or campaign or product relative);

information codification (e.g., techniques that associates information to a subject) or data summarization (e.g., cross data tabulations). The DBM development process concludes with *marketing knowledge*, which is the marketer interpretation of marketing information in actionable form. In this phase there has to be relevant information to support marketing activities decision.

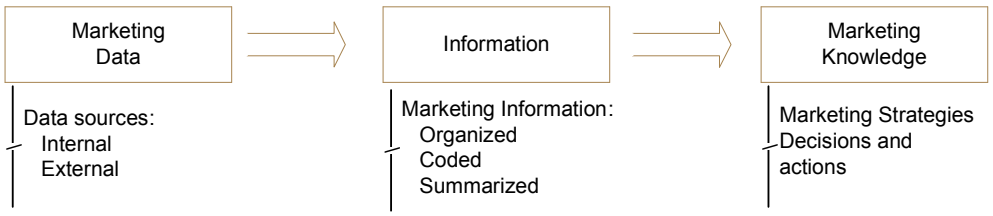


Figure 4: Database marketing general overall process

Technology based marketing is almost a marketing science imperative (Brookes *et al.*, 2004)(Zineldin and Vasicheva, 2008). As much as marketing research is improving and embracing new challenges its dependence on technology is also growing (Carson *et al.*, 2004). Currently, almost every organization has its own marketing information system, from single customer data records to huge data warehouses (Brito, 2000). Nowadays, DBM is one of the most well succeed marketing technology employment (Kurtulus and Kurtulus, 2006)(Frankland, 2007)(Lin and Hong, 2008)(Pinto *et al.*, 2009).

2.2.3 DBM Process with KDD

Database marketing is a capacious term related to the way of thinking and acting which contains the application of tools and methods in studies, their structure and internal organization so that they could achieve success on a fluctuating and difficult to predict consumer market (Lixiang, 2001).

For the present purpose we assume that, database marketing can be defined as a method of analyzing customer data to look for hidden, useful and actionable knowledge for marketing purposes. To do so, several different problem specifications may be referred.

These include market segmentation (Brito *et al.*, 2004), cross-sell prediction, response modeling, customer valuation (Brito and Hammond, 2007) and market basket analysis (Buckinx and den Poel, 2005) (Burez and Poel, 2007). Building successful solutions for these tasks requires the application of advanced DM and machine learning techniques to obtain relationships and patterns in marketing databases data and using this knowledge to predict each prospect's reaction to future situations.

In literature there are some examples about KDD usage in DBM projects usage for customers' response modeling whereas the goal was to use past transaction data of customers, personal characteristics and their response behavior to determine whether these clients were good or not (Coviello and Brodie, 1998) e.g., for mailing prospects during the next period (Pearce *et al.*, 2002) (den Poel and Buckinx, 2005). At these examples different analytical approaches were used: statistical techniques (e.g., discriminate analysis, logistic regression, CART and CHAID), machine learning methods (e.g., C4.5, SOM) mathematical programming (e.g., linear programming classification) and neural networks to model this customer's response problem.

Other KDD related application in DBM projects is customer retention activities. The retention of its customers is very important for a commercial entity, e.g., a bank or a oil distribution company. Whenever a client decides to change to another company, it usually implies some financial losses for this organization. Therefore, organizations are very interested in identifying some mechanisms behind such decisions and determining which clients are about to leave them. As an example one approach to find such potential customers is to analyze the historical data which describe customer behavior in the past (den Poel and Buckinx, 2005)(Buckinx and den Poel, 2005) (Rebelo *et al.*, 2006)(Burez and Poel, 2007)(Buckinx *et al.*, 2007).

2.3 Ontologies

Currently we live at a web-based information society. Such society has a high-level automatic data processing which requires a machine-understandable of representation of information's semantics. This semantics need is not provided by HTML or XML-based languages themselves. Ontologies fill the gap, providing a sharable structure and semantics of a given domain, and therefore they play a key role in such research areas such as knowledge management, electronic commerce, decision support or agent communication (Ceccaroni, 2001).

Ontologies are used to study the existence of all kinds of entities (abstract or concrete) that constitute the world (Sowa, 2000). Ontologies use the existential quantifier $\exists$ as a notation for asserting that something exists, in contrast to logic vocabulary, which doesn't have vocabulary for describing the things that exist.

They are also used for data-source integration in global information systems and for in-house communication. In recent years, there has been a considerable progress in developing the conceptual bases for building ontologies. They allow reuse and sharing of knowledge components, and are, in general, concerned with static domain-knowledge.

Ontologies can be used as complementary reusable components to construct knowledge-based systems (van Heijst *et al.*, 1997). Moreover, ontologies provide a shared and common understanding of a domain and describe the reasoning process of a knowledge-based system, in a domain and independent implementation fashion.

2.3.1 Ontology definitions

From the philosophy perspective, ontology is the theory or study of being, i.e., of the basic characteristics of all reality. Though the term was first coined in the 17th century, ontology is synonymous with metaphysics or first philosophy as defined by Aristotle in the 4th century BC (Guarino, 1995). Ontology is a part of metaphysics (Newell and level,

1982): it is the science of the existence which investigates the structure of being in general, rather than analyzing the characteristics of particular beings.

To answer the question "*but what is being?*" it was proposed a famous criterion but which did not say anything about what actually exists:"*To be is to be the value of a quantified variable*"(Quine, 1992). Those who object to it would prefer some guidelines for the kinds of legal statements. In general, further analysis is necessary to give the knowledge engineer some guidelines about what to say and how to say it.

From artificial intelligence literature there is a wide range of different definitions of the term ontology. Each community seems to adopt its own interpretation according to the use and purposes that the ontologies are intended to serve within that community. The following list enumerates some of the most important contributions:

- One of the early definitions is: 'An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.' (Neches *et al.*, 1991);
- A widely used definition is: 'An ontology is an explicit specification of a conceptualization.' (Gruber, 1993);
- An analysis of a number of interpretations of the word ontology (as an informal conceptual system, as a formal semantic account, as a specification of a conceptualization, as a representation of a conceptual system via a logical theory, as the vocabulary used by a logical theory and as a specification of a logical theory) and a clarification of the terminology used by several other authors is in Guarino and Giaretta work (Guarino, 1995).
- From Gruber's definition and more elaborated is: 'Ontologies are defined as a formal specification of a shared conceptualization.'(Borst *et al.*, 1997);
- 'An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base.' (Swartout *et al.*, 1996);
- A definition with an explanation of the terms also used in early definitions, states: 'conceptualization refers to an abstract model of some phenomenon in

the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared refers to the notion that an ontology captures consensual knowledge, that is, it is not primitive to some individual, but accepted by a group (Staab and Studer, 2004);

- An interesting working definition is: Ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and explicitly designates how concepts are interrelated which collectively impose a structure on the domain and constrain the possible interpretations of terms. Moreover, ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between communities. Such agreement facilitates accurate and effective communication of meaning, which in turn, leads to other benefits such as inter-operability, reuse and sharing. (Jasper and Uschold, 1999);
- More recently, a broad definition has been given: 'ontologies to be domain theories that specify a domain-specific vocabulary of entities, classes, properties, predicates, and functions, and to be a set of relationships that necessarily hold among those vocabulary terms. Ontologies provide a vocabulary for representing knowledge about a domain and for describing specific situations in a domain.' (Farquhar *et al.*, 1997) (Smith and Farquhar, 2008).

For this research, we have adopted as ontology definition: *A formal and explicit specification of a shared conceptualization, which is usable by a system in actionable forms*. Conceptualization refers to an abstract model of some phenomenon in some world, obtained by the identification of the relevant concepts of that phenomenon. Shared reflects the fact that an ontology captures consensual knowledge and is accepted by a relevant part of the scientific community. Formal refers to the fact that ontology is an abstract, theoretical organization of terms and relationships that is used as a tool for the analysis of the concepts of a domain. Explicit refers to the type of concepts used and the constraints on their use (Gruber, 1993)(Jurisica *et al.*, 1999). Therefore, ontology

provides a set of well-founded constructs that can be leveraged to build meaningful higher level knowledge. Hence, we consider that ontology is usable through systems in order to accomplish our objective: assistance work throughout actionable forms.

2.3.2 Reasons to use ontologies

Ontology building deals with modeling the world with shareable knowledge structures (Gruber, 1993). With the emergence of the Semantic Web, the development of ontologies and ontology integration has become very important (Fox and Gruninger, 1997) (Guarino, 1998) (Berners-Lee *et al.*, 2001). The SemanticWeb is a vision, for a next generation Web and is described in a Figure 7 called the "layer cake" of the Semantic Web (Berners-Lee, 2003) and presented in the Ontology languages section.

The current Web has shown that string matching by itself is often not sufficient for finding specific concepts. Rather, special programs are needed to search the Web for the concepts specified by a user. Such programs, which are activated once and traverse the Web without further supervision, are called agent programs (Zhou *et al.*, 2006). Successful agent programs will search for concepts as opposed to words. Due to the well known homonym and synonym problems, it is difficult to select from among different concepts expressed by the same word (e.g., Jaguar the animal, or Jaguar the car). However, having additional information about a concept, such as which concepts are related to it, makes it easier to solve this matching problem. For example, if that Jaguar *IS-A car* is desired, then the agent knows which of the meanings to look for.

Ontologies provide a repository of this kind of relationship information. To make the creation of the Semantic Web easier, Web page authors will derive the terms of their pages from existing ontologies, or develop new ontologies for the Semantic Web.

Many technical problems remain for ontology developers, e.g. scalability. Yet, it is obvious that the Semantic Web will never become a reality if ontologies cannot be

developed to the point of functionality, availability and reliability comparable to the existing components of the Web (Blanco *et al.*, 2008) (Cardoso and Lytras, 2009).

Some ontologies are used to represent the general world or word knowledge. Other ontologies have been used in a number of specialized areas, such as, medicine(Jurisica *et al.*, 1999)(CeSpivova *et al.*, 2004)(Perez-Rey *et al.*, 2006)(Kasabov *et al.*, 2007), engineering (Tudorache, 2006) (Weng and Chang, 2008), knowledge management (Welty and Murdock, 2006) (Diamantini *et al.*, 2006b), or business (Borges *et al.*, 2009) (Cheng *et al.*, 2009).

Ontologies have been playing an important role in knowledge sharing and reuse and are useful for (Noy and McGuinness, 2003):

- *Sharing common understanding* of the structure of information among people or software agents is one of the more common goals in developing ontologies (Gruber, 1993), e.g., when several different Web sites contain marketing information or provide tools and techniques for marketing activities. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications;
- *Enabling reuse of domain knowledge* was one of the driving forces behind recent surge in ontology research, e.g., models for many different domains need to represent the value. This representation includes social classes, income scales among others. If one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain;
- Making *explicit domain assumptions* underlying an implementation makes it possible to change these programming-language codes making these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users who must learn what terms in the domain mean;

- *Separating the domain knowledge from the operational knowledge* is another common use of ontologies, e.g., regarding computers hardware components, it is possible to describe a task of configuring a product from its components according to a required specification and implement a program that does this configuration independent of the products and components themselves. Then, it is possible develop an ontology of *PCcomponents* and apply the algorithm to configure made-to-order PCs. We can also use the same algorithm to configure elevators if we "feed" it an elevator component ontology (Rothenfluh *et al.*, 1996);
- *Analyzing domain knowledge* is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending them (Baader *et al.*, 2003).

Often ontology of the domain is not a goal in itself (Knublauch *et al.*, 2004a). Developing an ontology is akin to defining a set of data and their structure for other programs to use. Problem-solving methods, domain-independent applications, and software agents use ontologies and knowledge bases built from ontologies as data (van Heijst *et al.*, 1997) (Gottgtroy *et al.*, 2004) (Engelbach *et al.*, 2006). Within this work we have develop an DBM ontology and appropriate KDD combinations of tasks and tools with expected marketing results. This ontology can then be used as a basis for some applications in a suite of marketing-managing tools: One application could create marketing activities suggestions for data analyst or answer queries of the marketing practitioners. Another application could analyze an inventory list of a data used and suggest which marketing activities could be developed with such available resource.

2.3.3 Ontologies concepts

Here we use ontologies to provide the shared and common domain structures which are required for semantic integration of information sources. Even if it is still difficult to find consensus among ontology developers and users, some agreement about protocols, languages and frameworks exists. In this section we clarify the terminology which we will use throughout the thesis:

- *Axioms* are the elements which permit the detailed modeling of the domain. There are two kinds of axioms that are important for this thesis: defining axioms and related axioms. Defining axioms are defined as relations multi valued (as opposed to a function) that maps any object in the domain of discourse to sentence related to that object. A defining axiom for a constant (e.g., a symbol) is a sentence that helps defining the constant. An object is not necessarily a symbol. It is usually a class, or relation or instance of a class. If not otherwise specified, with the term axiom we refer to a related axiom;
- A *class* or *type* is a set of objects. Each one of the objects in a class is said to be an instance of the class. In some frameworks an object can be an instance of multiple classes. A class can be an instance of another class. A class which has instances that are themselves classes is called a meta-class. The top classes employed by a well developed ontology derive from the root class object, or thing, and they themselves are objects, or things. Each of them corresponds to the traditional concept of being or entity. A class, or concept in description logic, can be defined intentionally in terms of descriptions that specify the properties that objects must satisfy to belong to the class. These descriptions are expressed using a language that allows the construction of composite descriptions, including restrictions on the binary relationships connecting objects. A class can also be defined extensionally by enumerating its instances. Classes are the basis of knowledge representation in ontologies. Class hierarchies might be represented by a tree: branches represent classes and the leaves represent individuals.
- *Individuals*: objects that are not classes. Thus, the domain of discourse consists of individuals and classes, which are generically referred to as objects.

Individuals are objects which cannot be divided without losing their structural and functional characteristics. They are grouped into classes and have slots. Even concepts like group or process can be individuals of some class.

- *Inheritance* through the class hierarchy means that the value of a slot for an individual or class can be inherited from its super class.
- *Unique identifier*: every class and every individual has a unique identifier, or name. The name may be a string or an integer and is not intended to be human readable. Following the assumption of anti-atomicity, objects, or entities are always complex objects. This assumption entails a number of important consequences. The only one concerning this thesis is that every object is a whole with parts (both as components and as functional parts). Additionally, because whatever exists in space-time has temporal and spatial extension, processes and objects are equivalent.
- *Relationships:* relations that operate among the various objects populating an ontology. In fact, it could be said that the glue of any articulated ontology is provided by the network of dependency of relations among its objects. The class-membership relation that holds between an instance and a class is a binary relation that maps objects to classes. The *type-of* relation is defined as the inverse of *instance-of* relation. If *A* is an *instance-of B*, then *B* is a *type-of A*. The *subclass-of* (or *is-a*) relation for classes is defined in terms of the relation *instance-of*, as follows: a class *C* is a *subclass-of* class *T* if and only if all instances of *C* are also instances of *T*. The *superclass-of* relation is defined as the inverse of the *subclass-of* relation.
- *Role*: different users or any single user may define multiple ontologies within a single domain, representing different aspects of the domain or different tasks that might be carried out within it. Each of these ontologies is known as a role. In our approach we do not need to use roles since we only deal with a single ontology. Roles can be shared, or they can be represented separately in approaches without integration facilities. Moreover, roles can overlap in the sense that the same individuals can be classified in many different roles, but the class membership of an individual, its inherited slots and the values of those

slots may vary from role to role. A representation of the similarities and differences between two or more roles is known as a comparison.

- *Slots* (values that properties can assume). Objects have associated with them a set of own slots and each own slot of an object has associated with it a set of objects called slot values. Slots can hold many different kinds of values and can hold many at the same time. They are used to store information, such as name and description, which uniquely define a class or an individual. Classes have associated with them a collection of template slots that describe own slot values considered to hold for each instance of the class. The values of template slots are said to inherit to the subclasses and to the instances of a class. The values of a template slot are inherited to subclasses as values of the same template slot and to instances as values of the corresponding own slot. For example, the assertion that the gender of all *female* persons is *female* could be represented by the template slot *Gender* of class *Female-Person* having the value *Female*. If we create an instance of *Female-Person* called *Linda*, then *Female* would be the value of the own slot *Gender* of *Linda*. Own slots of an object have associated with them a set of own facets, and each own facet of a slot of a frame has associated with it a set of objects called facet values, e.g., the assertion that *Francisco* favorite foods must be *sweet food* can be represented by the facet Value-Type of the *Favorite-Food* slot of the *Francisco* frame having the value *Sweet-Food*. Template slots of a class have associated with them a collection of template facets that describe own facet values considered to hold for the corresponding own slot of each instance of the class. As with the values of template slots, the values of template facets are said to inherit to the subclasses and instances of a class. Thus, the values of a template facet are inherited to subclasses as values of the same template facet and to instances as values of the corresponding own facet.
- A *taxonomy* is a set of concepts, which are arranged hierarchically. A taxonomy does not define attributes of these concepts. It usually defines only the is-a relationship between the concepts. In addition to the basic is-a relation, the part-of relation may also be used;

- A *type* is an ontological category in artificial intelligence (in which it is synonymous of class) and in logic;
- A *vocabulary* is a language dependent set of words with explanations/documentation. It seeks universality and formality in a local context (for example a marketing domain).

Focusing on ontology reuse capability (one of the most important aspect in many research projects), we attain to assist the end user in new DBM and KDD projects through knowledge base instantiation and inference.

2.3.4 Methodologies to build ontologies

Although large-scale ontologies already exist, ontology engineers are still needed to construct the ontology for a particular task or domain, and to maintain and update the ontology to keep it relevant and up-to-date (Lopez *et al.*, 1999).

Ontology can be created from scratch, from existing ontologies only, from a body of information sources only; or a combination of the last two approaches. Ontological engineering is still a fairly immature discipline (Shen and Chuang, 2009) and several research groups propose various methods more commonly known as methodologies for building ontologies. There is no consensus between these groups and each employs its own methodology. Consequently some of the most current and popular methodologies used in ontology development will be described and discussed (Table 2).

Table 2: methodology comparison

Criteria	Methodology				
	Enterprise	**TOVE**	**Methontology**	**KACTUS**	**101**
Methodology Details	Little	Little	A lot	Little	Regular
Formalization recommendation	None	Logic	None	None	None
Applications building strategy	Independent	Semi dependent	Independent	Dependent	Semi dependent
Concepts identifying strategy	Middle-out	Middle-out	Middle-out	Top-down	Middle-out
Recommended lifecycle	None	None	Yes	None	Yes

Since ontologies are part (sometimes only potentially) of software products, we use the IEEE standard software development processes[1], to introduce some metrics regarding methodology performance (Lopez, 1999):

- *Methodology details*: concerning of whether the activities and techniques proposed by the methodology are exactly specified or not;
- *Recommendations for knowledge formalization*: evaluation of the formalism or formalisms proposed for representing knowledge (logic, frames, etc.);
- *Strategy for ontologies construction*: Discussion of which of the following strategies are used to develop ontologies:
  - Application-dependent: the ontology is built on the basis of an application knowledge base, by means of a process of abstraction.
  - Application semi-dependent: possible scenarios of ontology use are identified in the specification stage.
  - Application-independent: the process is totally independent of the uses to which the ontology will be put in knowledge-based systems, agents, etc.
- *Strategy for concepts identification*: The possible strategies are (Jasper and Uschold, 1999): from the most concrete to the most abstract (bottom-up), from the most abstract to the most concrete (top-down), or from the most relevant to the most abstract and most concrete (middle-out);

[1] IEEE 1074-2006 - IEEE Standard for Developing a Software Project Life Cycle Process

- *Recommended life cycle*: how does the methodology implicitly or explicitly propose a life cycle. The methodology should recommend one or more life cycles from which the developer can select one.

Analysis of methodologies summary

According to the above analysis, presented in

Table 2, it is possible to shape that none of the methodologies are fully mature when compared with the IEEE standard. Analyzed proposals are not unified. Nowadays each group applies its own methodology. Therefore, efforts are required along the lines of unifying methodologies to arrive at a situation resembling Knowledge and Software Engineering (Gomez-Perez *et al.*, 2004). Although the following scale can be established:

i. Methontology is the most mature; however, recommendations for the pre-development processes are needed, and some activities and techniques should be specified in more detail (as recommended by FIPA[2]).
ii. Enterprise methodology, is confined to the business domain: this methodology has to few details and does not include any activities description or life cycle;
iii. Kactus methodology, has the same above omissions. We register that it has an application dependent strategy and a top-down strategy for concepts identifying;
iv. TOVE methodology has the same omissions as the above methodology. However, it has a logic formalization recommendation;

Nevertheless, attempts to unify two or more methodologies are already in progress (Lovrencic and Cubrilo, 2007)(Blanco *et al.*, 2008)(Borges *et al.*, 2009). We have found a series of methodologies that can be used as a reference point for developing one or several standardized methodologies adaptable to different ontology types in different settings.

Next, we present the aforementioned methodologies for ontologies development.

[2] FIPA – Foundation for Intelligent Physical Agents

Enterprise Methodology

This methodology is based on the experience of developing the Enterprise Ontology, an ontology for enterprise modeling processes (Uschold and King, 1995). This methodology provides guidelines for developing ontologies, which are (Jasper and Uschold, 1999) (Uschold and Gruninger, 2004):

- *Domain Knowledge capture*. It is important to be clear why the ontology is being built and what its intended uses are. This task may be broken into three steps (Fox and Gruninger, 1998):
  - Identification of the key concepts and relationships in the domain of interest, that is, scoping. It is important to centre on the concepts as such, rather than the words representing them;
  - Production of precise unambiguous text definitions for such concepts and relationships; and,
  - Identification of terms to refer to such concepts and relationships.
- *Coding*. Involves explicitly representing the knowledge acquired in the previous step in a formal language;
- *Integrating existing ontologies*. During either or both of the capture and coding processes, there is the question of how and whether to use ontologies that already exist;
- *Evaluation*, to make a technical judgment of the ontologies, their associated software environment, and documentation with respect to a frame of reference (requirements specifications, competency questions, and/or the real world) (Gomez-Perez *et al.*, 2004);
- *Documentation* recommends that guidelines be established for documenting ontologies, possibly differing according to the type and purpose of the ontology.

Enterprise methodology evaluation

According to the criteria referred previously, the following can be said:

- *Detail of the methodology*. This methodology does not precisely describe the techniques and activities, therefore it could be considered with low detail level.
- *Recommendations for knowledge formalization*: None in particular.
- *Strategy for applications development:* The process is totally independent of the uses to which the ontology will be put and is, therefore, application independent
- *Strategy for concepts identification*: key concepts are established by searching first for the most important, rather than the most general or most particular concepts; the others are obtained by generalization and by specialization. Therefore, a middle-out strategy can be said to be used for identifying concepts.
- *Recommended life cycle*. This methodology does not propose a life cycle, however, it proposes some structured like, requirements (e.g., environment study), implementation (e.g., feasibility study) and integral processes (e.g., training and configuration management).

TOVE Methodology

This methodology is based on the experience in developing the TOVE project ontology (Fox and Gruninger, 1997) within the domain of business processes and activities modeling. Briefly, it involves building a logical model of the knowledge that is to be specified by means of the ontology. This model is not constructed directly. First, an informal description is made of the specifications to be met by the ontology and then this description is formalized.

TOVE methodology proposed steps are as follows (Fox and Gruninger, 1997) (Fox and Gruninger, 1998):

- *Motivating scenarios capture*: the development of ontologies is motivated by scenarios that arise in the application (Fox and Gruninger, 1997). The motivating scenarios are story problems or examples which are not adequately addressed by existing ontologies. Moreover, a motivating scenario provides a set of intuitively possible solutions to the scenario problems. These solutions provide an informal

intended semantics for the objects and relations that will later be included in the ontology. Any proposal for a new ontology or extension to an ontology should describe one or more motivating scenarios, and the set of intended solutions of problems presented in the scenarios;

– *Informal competency questions*: These are based on the scenarios obtained in the preceding step and can be considered as expressiveness requirements that are in form of questions. The ontology must be able to represent these questions using its terminology, and be able to characterize the answers to these questions using the axioms and definitions. The competency questions are stratified and the response to one question can be used by means of composition and decomposition operations, to answer more general questions from the same or another ontology. Therefore, this is a means of identifying knowledge already represented for reuse and integrating ontologies;

The questions serve as constraints on what the ontology can be, rather than determining a particular design with its corresponding ontological commitments. Instead, the competency questions are used to evaluate the ontological commitments that have been made to see whether the ontology meets the requirements or not.

TOVE methodology evaluation

According to previous defined criteria, the following can be said:

- *Methodology details*. Neither the activities nor the techniques are described in detail.
- *Knowledge formalization*. Clearly opts for logic.
- *Strategy for applications building*: Ontology use scenarios are identified in the specification stage, so it is a application-semi dependent strategy.
- *Strategy for identifying concepts*: It adopts a middle-out strategy.
- *Life cycle*. No life cycle mode selection process is identified, nor is any explicit reference made to there being any preference for one model over another; however, there is a defined order in which the development activities are performed. Also a

provision is made for extending an ontology that has already been built, starting again with getting scenarios.

KACTUS Methodology

Kactus methodology is born from the Esprit Kactus Project (Bernaras *et al.*, 1996). One of the objectives of this project is to investigate the feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it. However, that approach to developing ontologies is conditioned by applications development. That is, every time an application is built, the ontology that represents the knowledge required for the application must to be also, built. This ontology can be developed by reusing others and can also be integrated into the ontologies of later applications. Therefore, every time an application is developed, the following steps are taken (Bernaras *et al.*, 1996):

- *Application specification*, which provides an application context and a view of the components that the application tries to model;
- *Preliminary design based on relevant top-level ontological categories*, where the list of terms and tasks developed during the previous phase is used as input for obtaining several views of the global model in accordance with the top-level ontological categories determined. This design process involves searching ontologies developed for other applications, which are refined and extended for use in the new application;
- *Ontology refinement and structuring* in order to achieve a definitive design. The principles of minimum coupling can be used to assure that the modules are not very dependent on each other and are as coherent as possible, looking to get maximum homogeneity within each module.

Kactus methodology evaluation

From past criteria, the following can be said:

- *Methodology details*: Very little.

- *Knowledge formalization*: None.
- *Strategy for building applications*: The construction of ontologies is based on the construction of particular applications. As more applications are built, the ontology becomes more general, and, therefore, moves further away from what would be a traditional knowledge base. So, this methodology can be said to follow an application-dependent strategy in this respect.
- *Strategy for identifying concepts*: KACTUS uses Top-down approach for concepts identification.
- *Life cycle*. It simply seems to assume that the life cycle should be the same as is used in the development of the application associated with the ontology.

Methontology methodology

This methodology was developed within the Laboratory of Artificial Intelligence at the Polytechnic University of Madrid. The methontology framework (Fernandez *et al.*, 1997) (Blazquez *et al.*, 1998) (Gomez-Perez *et al.*, 2004) enables the construction of ontologies at the knowledge level and includes (Blazquez *et al.*, 1998): the identification of the ontology development process, a life cycle based on evolving prototypes, and particular techniques for carrying out each activity.

Ontology Development Process

The ontology development process refers to which activities are carried out when building ontologies (Fernandez *et al.*, 1997). Methontology identifies three main types of activities: Project management; Development-oriented activities and support activities. We briefly describe them.

- *Project management* related activities include (Fernandez *et al.*, 1997) (Lopez, 1999) (Lopez *et al.*, 1999):
  - Planning, identifies which tasks are to be performed, how they will be arranged, how much time and what resources are needed for their completion. This activity is essential

- for ontologies that need to use ontologies which have already been built or ontologies that require levels of abstraction and generality;
- Control, guarantees that planned tasks are completed in the manner that they were intended to be performed;
- Quality assurance: assures that the quality of each and every product outputted (ontology, software and documentation) is satisfactory, describing how these activities are performed.

- *Development-oriented related activities* include (Fernandez *et al.*, 1997):
  - Specification: states why the ontology is being built and what are its intended uses and who are the end-users;
  - Conceptualization structures the domain knowledge as meaningful models at the knowledge level;
  - Formalization: transforms the conceptual model into a formal or semi-computable model.
  - Implementation: builds computable models in a computational language;
  - Maintenance: updates and corrects the ontology.

- *Support related activities* include a series of activities, performed at the same time as development-oriented activities, without which the ontology could not be built, as the following (Fernandez *et al.*, 1997)(Lopez *et al.*, 1999) (Gomez-Perez *et al.*, 2004):
  - Knowledge Acquisition acquires knowledge of a given domain;
  - Evaluation makes a technical judgment of the ontologies, their associated software environments and documentation with respect to a frame of reference during each phase and between phases of their life cycle;
  - Integration of ontologies is required when building a new ontology reusing other ontologies that are already available;
  - Documentation details, clearly and exhaustively, each and every one of the phases completed and products generated;

There are remarkable works developed according to methontology: Chemicals (Blazquez *et al.*, 1998) (Gomez-Perez *et al.*, 2004) which contains knowledge within the domain of chemical elements and crystalline structures; Environmental pollutants ontologies (Gomez-Perez and Rojas-Amaya, 1999), used to represent the methods of detecting the different pollutant components of various media (e.g., water, air or soil); The Reference-Ontology (Arpirez *et al.*, 2000) - an ontology that plays the role of a kind of yellow pages of ontologies. It gathers, describes and has links to existing ontologies, using a common logical organization. Also, this methodology has been proposed for ontology construction by the Foundation for Intelligent Physical Agents (FIPA[3]), which promotes inter-operability across agent-based applications.

Methontology Life Cycle

Ontology development cycle is the most time stable approach to conceive and understand the process that aims to produce an ontology. The usually accepted main phases through which an ontology is built are knowledge acquisition, evaluation and documentation (Figure 5).

- *Knowledge acquisition:* refers to the acquire knowledge about the subject either by using elicitation techniques on domain experts or by referring to relevant bibliography. Several techniques can be used to acquire knowledge, such as brainstorming, interviews, questionnaires, text analysis, and inductive techniques;
- *Evaluation:* relate all activities concerning with ontology operation. Technically judge the quality of the ontology;
- *Documentation:* register and report what was done, how it was done and why it was done. Documentation associated with the terms represented in the ontology is particularly important, not only to improve its clarity, but also to facilitate maintenance, use and reuse

[3] http://www.fipa.org

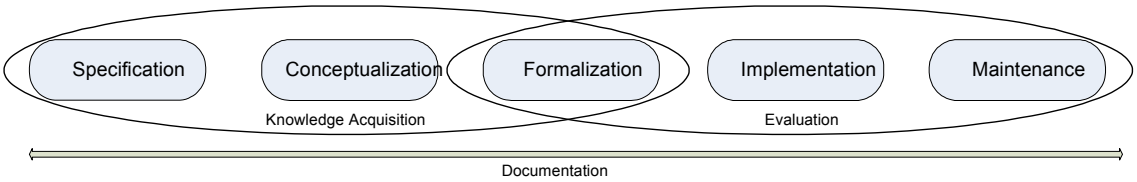


Figure 5: Activities in the ontology development life cycle (adapted from (Pinto and Martins, 2004)).

Each one the refer phases may include different activities, like specification, conceptualization, formalization, implementation and maintenance (Pinto and Martins, 2004):

- *Specification:* Identifies the purpose and scope of the ontology. Purposes answers the questions like "Why is the ontology being built?" and scopes answers to the question "What are its intended uses and end users?"
- *Conceptualization:* Represents, in a conceptual model, the ontology to be built, so that it meets the specification found in the previous step. There are different conceptual models propose in different methodologies (Ekes *et al.*, 1997)(Lopez, 1999)(Jarrar, 2005). The conceptual model of ontology consists of concepts in the domain and relationships among those concepts. Relationships enhance stronger connections between groups of concepts. These groups of highly connected concepts usually correspond to different modules into which the domain can be decomposed.
- *Formalization:* This activity transforms the conceptual description into a formal model, that is, the description of the domain found in the previous step is written in a more formal form, although not yet its final form. Concepts are usually defined through axioms that restrict the possible interpretations for the meaning of those concepts. Concepts are usually hierarchically organized through a structuring relation, such as *is-a* (*class-superclass*, *instance-class*) or *part-of* (*belongs to*);
- *Implementation:* Implement the formalized ontology in a knowledge representation language. For that, one commits to a representation ontology,

chooses a representation language and writes the formal model in the representation language using the representation ontology;

- *Maintenance:* Update and correct the implemented ontology. There are also activities that should be performed during the whole life cycle;

Methontology Evaluation

According to the criteria set out, the following can be said:

- *Detail of the methodology*. A sizable part of the methodology is very detailed the remainder will be specified in more detail in the future.
- *Knowledge formalization*. Methontology gives freedom of choice with regard to formalization;
- *Strategy for building applications*. Application independent;
- *Strategy for identifying concepts*. The most relevant concepts are identified first, so it adopts a middle-out strategy;
- *Life cycle*: methontology uses evolving prototypes.

101Methodology

101 Methodology is based on the principle that, there are several viable alternatives for ontology development and ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.

Ontologies have become core components of many large applications. This methodology presents a set of tasks for creating ontologies based on declarative knowledge representation (Figure 6). It leverages the author's experiences developing and maintaining ontologies in a number of ontology environments including Protégé [4], Ontolingua [5], or Chimaera[6]. The Ontology 101 methodology is relatively simple, since

[4] http://protege.stanford.edu/
[5] http://www.ksl.stanford.edu/software/ontolingua/
[6] http://www-ksl.stanford.edu/software/chimaera/

it defines simple and, some of them, generic steps. Indeed this methodological approach does not make assumptions about knowledge representation/ontology language.

Figure 6: 101 methodology steps

This methodology uses the ontology domain and scope (based on related knowledge) to pragmatically determine which the best approach for ontology development is. Also, this methodology assumes the ontology development is a process of iterative design that will likely continue through the entire lifecycle of the ontology.

The ontology development process refers to which activities are carried out when building ontologies. 101 methodology uses a generic approach from the domain application to the effective instance creation, through the following steps (Noy and McGuinness, 2003):

- *Determining the domain and scope of the ontology*, through the answer to some basic but relevant questions, like what is the domain that the ontology will cover; what is it going to be used for; or for what type of questions the information in the ontology should provide answers. Moreover, one of the ways to determine the scope of the ontology is through competency questions (Fox and Gruninger, 1997). Questions like, does the ontology contain enough information to provide the aforementioned answers; or, do the answers require a particular level of detail or representation of a particular area. These competency questions are just a sketch and do not need to be exhaustive;
- *Considering reusing existing ontologies*: since it is almost always worth considering what someone else has done and checking if we can refine and extend existing sources for some particular domain and task. Reusing existing ontologies may be a requirement if the system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies. The formalism in which an ontology is expressed often does not matter, since many knowledge representation systems can import and export ontologies;

- *Enumerating important terms* in the ontology throughout an extensive list of all domain related knowledge terms, with which it would be useful either to make statements about or to explain to a user. As an example, important marketing-related terms will include, client, customer or marketing activity; different types of data, such as personal, market or financial; subtypes of personal data information type such as psychographics, demographics, life style or transactional;
- *Defining the classes and the class hierarchy*: There are several possible approaches in developing a class hierarchy (Uschold and King, 1995) (Jasper and Uschold, 1999): A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts; Bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts; and, A combination development process is a combination of the top-down and bottom-up approaches: firstly some more salient concepts are identified and then they are generalize and specialize them appropriately. None of these three methods is inherently better than any of the others. The approach to take depends strongly on the personal view of the domain. If a developer has a systematic top-down view of the domain, then it may be easier to use the top-down approach. The combination approach is often the easiest for many ontology developers, since the concepts "in the middle" tend to be the more descriptive concepts in the domain.
- *Properties definition*: classes alone will not provide enough information to answer the competency questions arisen step 1. Once some classes have been defined, the internal concept structure – properties must be identified. These terms include, as example, customer's birth date, gender or address post code. Each property in the list, it must determine which class it describes. These properties become slots attached to classes. Thus, the Client class will have the following slots: address post code, gender and birth date; and, the class Data will have an information type slot. In general, there are several types of object properties that can become slots in an ontology: "intrinsic" properties such as the client gender; "extrinsic" properties such as a client's address, and area it comes from; "parts", if the object

is structured; these can be both physical and abstract "parts" (e.g., client data and classification); and, relationships with other individuals - these are the relationships between individual members of the class and other items (e.g., the data categorizer, representing a relationship between a data task, data items and new form of data: categorized output). All subclasses of a class inherit the slot of that class, e.g., all slots of the class Data will be inherited to all subclasses of Information Data Type , including Source and Structure Type;

- *Defining the facets of the slots (restrictions)*: slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take, e.g., the value of a name slot (as in "client name") is one string. That is, name is a slot with value type String. A slot data categorizer (operational data task) can have multiple values and the values are instances of the class Data. That is, categorizer is a slot with value type *Instance* with *Data* as allowed class. There are several common facets:
  - Slot cardinality (defines how many values a slot can have);
  - Slot-value type (A value-type facet describes what types of values can fill in the slot, e.g., string, number or date);
  - Domain (The classes to which a slot is attached or a classes which property a slot describes); and
  - Range of a slot (allowed classes for slots of type *instance*);

- *Creating instances*: this last step consists in creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (i) choosing a class, (ii) creating an individual instance of that class, and (iii) filling in the slot values.

101 Methodology evaluation

- *Methodology details:* Regular;
- *Knowledge formalization:* None;
- *Strategy for applications building*: some usage scenarios are identified in the specification stage, so it is an application-semi dependent strategy;

- *Strategy for identifying concepts*: 101 uses middle-out approach for concepts identification. Key concepts are established by searching first for the most important, rather than the most general or most particular concepts; the others are obtained by generalization and by specialization;
- *Life cycle*, uses a set of well defined steps. Nevertheless, this methodology also assumes the interaction between each life style phase.

2.3.5 Ontology languages

An ontology language is a formal language by which an ontology is built (Lovrencic and Cubrilo, 2007)(Weng and Chang, 2008). Currently several ontology implementation languages exist and they can be divided into three main types (Gruber, 1993) (Nedellec and Nazarenko, 2005): vocabularies defined using natural language; object based-knowledge representation languages such as frames; and, UML, and languages based on first order predicate logic such as Description Logics.

The more classic ontology languages have been developed during the '90s include, KIF-based Ontolingua[7], LOOM[8] or Frame Logic (F-Logic[9]). The knowledge representation paradigm underlying these languages was based on first order logic [10] or a combination of frames and first order logic or on Description Logics[11] (e.g. LOOM). More recent, and latter explained, ontology implementation languages include; RDF, RDF Schema, XOL, SHOE, OIL, DAML+OIL and OWL. These languages are in a

[7] http://ontolingua.stanford.edu/
[8] http://www.isi.edu/isd/LOOM
[9] www.neon-toolkit.org/wiki/index.php/F-Logic_Modeling
[10] **First-order logic** is a formal logic used in mathematics, philosophy, linguistics, and computer science. It goes by many names, including: first-order predicate calculus, the lower predicate calculus, and predicate logic. First-order logic is distinguished from propositional logic by its use of quantifiers; each interpretation of first-order logic includes a domain of discourse over which the quantifiers range.
[11] **Description logics** (DL) are a family of knowledge representation languages which can be used to represent the concept definitions of an application domain in a structured and formally well-understood way. DL refers, to concept descriptions used to describe a domain and, to the logic-based semantics which can be given by a translation into first-order predicate logic.

constant state of evolution. As XML has emerged as a standard language to exchange information on the web also, they have been created based on XML to implement ontologies.

Classic Ontology Specification Languages

Ontolingua was created in the early 1990s to support the design and specification of ontologies with a clear logical semantics based on KIF (Neches *et al.*, 1991)(Gruber, 1993)(Jurisica *et al.*, 1999). The Ontolingua ontology development environment provides a suite of ontology authoring tools and a library of modular, reusable ontologies. The environment is available as a World Wide Web service and has a substantial user community. The tools in Ontolingua are oriented toward the authoring of ontologies by assembling and extending ontologies obtained from a library. Moreover Ontolingua has formalism set for combining the axioms, definitions, and words (non-logical symbols) of multiple ontologies (Ekes *et al.*, 1997).

LOOM[12] was developed at the same time as Ontolingua and is based on DL and provides automatic classifications. Loom is a language and environment for constructing intelligent applications. The heart of Loom is a knowledge representation system that is used to provide deductive support for the declarative portion of the Loom language. Declarative knowledge in LOOM consists of definitions, rules, facts, and default rules. A deductive engine denominated as classifier utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies in order to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing.

OCML[13] (Operational Conceptual Modeling Language) was developed later in 1993 (Motta, 1998) it is a frame based language and can be considered as a kind of operational Ontolingua because it provides deductive and production rules and function evaluation facilities for its constructs. The OCML modeling language supports the construction of

[12] http://www.isi.edu/isd/LOOM
[13] http://technologies.kmi.open.ac.uk/ocml

knowledge models by means of several types of constructs. It allows the specification and operationalization of functions, relations, classes, instances and rules. It also includes mechanisms for defining ontologies and problem solving methods, the main technologies developed in the knowledge modeling area.

F-Logic[14] was developed in 1995 and combines frames and first logic (Kifer *et al.*, 1995). Its inference engine Ontobroker (Decker *et al.*, 1998)can be used for constraint checking and deducting new information (Farquhar *et al.*, 1997). This language is usable to create a basic ontology by integrating existing information and the expert knowledge. An ontology engineer has to be formalized in a type of rules in creating ontologies. Moreover, F-Logic may be generally applied after the domain terms and properties identification.

Web-Based Ontology Specification Languages

The Internet has promoted the web based ontology languages creation (Horrocks *et al.*, 2005).The World Wide Web Consortium (W3C) recommends a number of standards as part of the Semantic Web stacks (Figure 7).

The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. At its core, the semantic web comprises a set of design principles, collaborative working groups, and a variety of enabling technologies.

[14] http://www.wsmo.org/2004/d16/d16.2/v0.1/20040324/

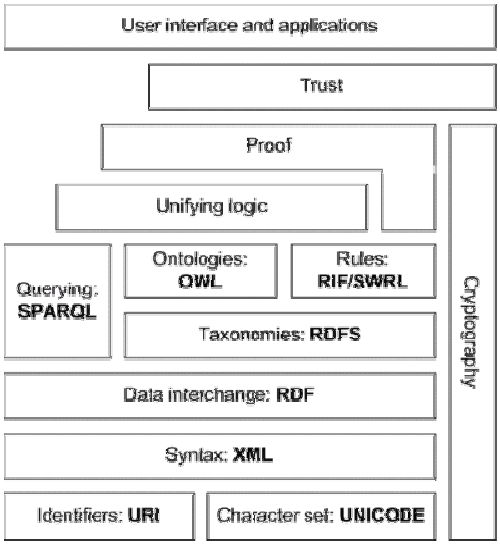


Figure 7: web stack (adapted from (Berners-Lee, 2003))

The semantic web comprises the standards and tools of XML, XML Schema, RDF, RDF Schema and OWL that are organized in the Semantic Web Stack. The OWL Web Ontology Language Overview describes the function and relationship of each of these components of the semantic web (Smith and Farquhar, 2008)(Berners-Lee *et al.*, 2001)(Berners-Lee, 2003):

- XML provides an elemental syntax for content structure within documents, yet associates no semantics with the meaning of the content contained within. XML Schema is a language for providing and restricting the structure and content of elements contained within XML documents.
- RDF is a simple language for expressing data models, which refer to objects ("resources") and their relationships. An RDF-based model can be represented in XML syntax. RDF schema was built as an extension to RDF, and the combination of RDF and RDF Schema is known as RDF(S). RDF Schema is a vocabulary for describing properties and classes of RDF-based resources, with semantics for generalized-hierarchies of such properties and classes;
- OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, properties characteristics (e.g. symmetry), and enumerated classes. OWL the Web ontology language is an emerging ontology language standard that has been optimized for data exchange and knowledge

sharing. OWL is the mainstream tool for modeling ontologies and was developed by the Web Ontology working group in 2001. It is used when information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL is a DL based language and can be used explicitly to represent the meaning of terms in vocabularies and the relationships between those terms.

- SPARQL (Simple Protocol and RDF Query Language) is a protocol and query language for semantic web data sources (Kalfoglou and Robertson, 2000) (Kalfoglou and Schorlemmer, 2007). An example of a SELECT query follows.

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name .
        ?x foaf:mbox ?mbox . }
```

The first line defines namespace prefix, the last two lines use the prefix to express a RDF graph to be matched. Identifiers beginning with question mark *?* identify variables. In this query, we are looking for resource *?x* participating in triples with predicates *foaf:name* and *foaf:mbox* and want the subjects of these triples (Kalfoglou and Schorlemmer, 2007);

- SWRL (Semantic Web Rule Language) is an expressive OWL-based rule language. SWRL allows users to write rules that can be expressed in terms of OWL concepts to provide more powerful deductive reasoning capabilities than OWL alone. Semantically, SWRL is built on the same description logic foundation as OWL and provides similar strong formal guarantees when performing inference.

  Generally, a SWRL rule contains an antecedent part, which is referred to as the body, and a consequent part, which is referred to as the head. Both the body and head consist of positive conjunctions of atoms. SWRL does not support negated atoms or disjunction. Informally, a SWRL rule may be read as meaning that if all the atoms in the antecedent are true, then the consequent must also be true. Also, SWRL rules are written in terms of OWL classes, properties, individuals, and data values.

As example, a SWRL rule expressing that a *person* with a *male* sibling *has a brother* would require capturing the concepts of *person*, *female*, sibling of and *brother of* in OWL. Intuitively, the concept of *person* and *male* can be captured using an OWL class called *Person* with a subclass *Man*; the sibling and brother relationships can be expressed using OWL object properties *hasSibling* and *hasBrother* with a domain and range of *Person*. The rule in SWRL would then be:

Person(?p) ^ hasSibling(?p,?s) ^ Man(?s) -> hasBrother(?p,?s)

Executing this rule would have the effect of adding the *hasBrother* property to all OWL individuals with one or more male siblings and assigning its value to those siblings. Similarly, a rule that asserts that all persons that own a car should be classified as drivers can be written as follows:

Person(?p) ^ hasCar(?p, true) -> Driver(?p)

Executing this rule would have the effect of classifying all car-owner individuals of type *Person* to also be members of the class *Driver*.

One of the most interesting aspects about SWRL is that it shares OWL's open world assumption, e.g., one might expect that a rule that infers that if two OWL individuals of type *Author* cooperate on the same *publication* that they are *collaborators* could be written as:

Publication(?p)^ hasAuthor(?p, ?y) ^ hasAuthor(?p, ?z) ->collaboratesWith(?y, ?z)

Another important SWRL powerful feature is its ability to support user-defined built-ins[15].An example SWRL rule using a core SWRL built-in to indicate that a customer with relevant transactions greater than 100€ is an VIP client is:

client(?p)^hasTransaction(?p,?tra)^swrlb:greaterThan(?tra,100)→ VIPclient(?p)

By convention, core SWRL built-ins are preceded by the namespace qualifier *swrlb*. When executed, this rule would classify individuals of class *Client* with an

[15] Built-in: A built-in is a predicate that takes one or more arguments and evaluates to true if the arguments satisfy the predicate. For example, an equal built-in can be defined to accept two arguments and return true if the arguments are the same. A number of core built-ins for common mathematical and string operations are contained in attach section SWRL Built-in.

hasTransaction property value of greater than 100 as members of the class *VIPclient*.

2.3.6 Ontology development Tools

Ontology development or engineering tools include suites and environments that can be used to build a new ontology from scratch or by reusing existing ontologies. Since the mid-nineties there has been an exponential increase in the development of technological platforms related with ontologies. Some of the older environments which are in a state of stable development include Ontosaurus, Ontolingua and WebOnto. More recent tools include; OntoEdit, WebODE, Protégé or KAON. A general description of these ontology development Tools is shown in Table 3.

Table 3: Ontology development tools overview

	Developers	Inference Engine	Availability	Import format	Export format	Graph view
Ontolingua	KMI – open university	No	Free web access	IDL, KIF	KIF, CLIPS, IDL, OKBC syntax, Prolog syntax	No
OntoSaurus	University of Southern California	No	Open source + Free evaluation	XML	OIL, XML,	
WebOnto	Open University	Yes	Free web access	OCML	OCML, GXL, RDF(S) and OIL	Yes
Protégé	Stanford University	Pellet FAC++	Freeware	XML, RDF(S), XML Schema	XML, RDF(S), XML Schema, FLogic, CLIPS, Java html	Via plug-ins like GraphViz and Jambalaya
KAON	University of Karlsruhe	Yes	Free web access	RDF(S)	RDF(S)	No
WebODE	University Madrid	Through prolog	Freeware	RDF(S), UML, DAML+OIL and OWL	RDF(S), UML, DAML+OIL,OWL, PROLOG, X-CARIN, Java/Jess	Form based graphical user interface
OntoEdit	Ontoprise	No	Freeweb access	XML, RDF(S), FLogic and DAML+OIL	XML, RDF(S), FLogic and DAML+OIL	Yes

The *Ontolingua* was the first ontology tool created (Gruber *et al.*, 1990) (Gomez-Perez *et al.*, 2004). Developed in the knowledge systems laboratory at Stanford University, it was built to ease the development of Ontolingua ontologies. Initially the main module inside the ontology server was the ontology editor and other modules like Webster (an equation solver) and OKBC (Open knowledge Based Connectivity);

OntoSaurus[16] was developed around the same time as Ontolingua by the Information Sciences Institute at the University of South California. OntoSaurus consists of two modules: an ontology server, which uses LOOM as its knowledge representation system and a web browser for LOOM ontologies. Ontologies developed through OntoSaurus can also be accessed with the OKBC protocol;

WebOnto is an ontology editor for OCML (Operational Conceptual Modeling Language) ontologies and was developed at the Knowledge Media Institute at Open University. This tool is a Java applet coupled with a customized web server and allows users to browse and edit knowledge models over the internet. The fact that WebOnto was able to support collaborative ontology editing was a major advantage at the time (Corcho *et al.*, 2003) (Staab and Studer, 2004).

The above environments described (Ontolingua, Ontosaurus and WebOnto) were created solely for browsing and editing ontologies purpose in a specific language (Ontolingua, LOOM and OCML respectively). They are the older generation of editors and they were hardly extensible compared to the engineering environments of today.
More recent generation of ontology engineering environments are more advanced and ambitious than their predecessors (Staab and Studer, 2004) (Nigro *et al.*, 2008). They possess highly extensible, component based architectures, where new modules can easily be added to provide more functionality to the environment.

Recent Generation Ontology development tools

[16] http://www.webkb.org/kb/ontology.html

WebODE[17] is an easily extensible and scalable ontology workbench developed by the Ontology Group at the Technical University of Madrid. It is the successor of the Ontology Design Environment (ODE). WebODE is used as a Web server with a Web Interface. The core of this environment is the ontology access service which is used by all the services and applications plugged into the server. The ontology editor also provides constraint checking capabilities, axiom rule creation and parsing with the WebODE Axiom Builder editor, documentation in HTML, ontology merge, and ontology exportation and importation in different formats (e.g., XML, OIL, F-logic, or Java). Its inference built in service uses Prolog and a subset of the OKBC protocol (Arpirez *et al.*, 2000) (Staab and Studer, 2004).

OntoEdi[18]*t* is developed by Artificial Intelligence laboratory of University of Karlsruhe and is built on top of a powerful internal ontology model. The internal ontology model can be serialized using XML, which supports the internal file handling. It supports F-Logic, RDF-Schema and OIL. In the current version OntoEdit has an interface to the F-Logic Inference Engine (the backbone of OntoBroker), in the next version the FaCT system will be accessible from OntoEdit. The tool is based on a flexible plug-in framework. Also exists the professional version of OntoEdit which contains several additional plug-ins, a collaborative environment and inference capabilities (Staab and Studer, 2004).

Protégé[19] is one of the most widely used editing tools and has been developed by the Stanford Medical Informatics group at Stanford University and the information management group at. The design and development of Protégé has been driven primarily by two goals: to be compatible with other systems for knowledge representation and to be an easy to use and configurable tool for knowledge extraction. It is an open source, standalone application with an extensible architecture, which assists users in the construction of large electronic knowledge bases. The core of this environment is an ontology editor. Numerous plug-ins provide several functions

[17] http://webode.dia.fi.upm.es/WebODEWeb
[18] www.ontoknowledge.org/tools/ontoedit
[19] http://Protege.stanford.edu

including alternative visualization mechanisms, management of multiple ontologies, inference services and ontology language importation/exportation. Protégé is a development environment for ontologies and knowledge-based systems.
The OWL plug-in is an extension of Protégé with support for the Web Ontology Language (OWL). The protégé OWL plug-in enables users to; load and save OWL and RDF ontologies, edit and visualize classes, properties and SWRL rules, define logical class characteristics as OWL expressions, execute reasoners such as description logic classifiers and finally to edit OWL individuals for Semantic Web markup (Knublauch *et al.*, 2004b).

KAON[20] is an open-source ontology management system targeted for business applications. It includes a comprehensive tool suite allowing easy ontology creation and management and provides a framework for building ontology-based applications (Staab and Studer, 2004). KAON provides two user-level applications: OiModeler (ontology editor and provides support for ontology creation and maintenance) and KAON portal (provides a simple framework for navigating and searching ontology's through Web browsers).
KAON is primarily a framework for the development of other ontology-based applications. It has the following modules: Front-end (the front-end is mainly presented by two applications, OI-modeler and KAON Portal); KAON Core (the core of KAON is the two APIs for the RDF and the KAON ontology language); and, KAON Libraries (to support and provide all functionalities).

2.3.7 Inference

Description Logics (DLs) are a family of logic based knowledge representation formalisms (Baader *et al.*, 2003). Although they have a range of applications, they are perhaps best known as the basis for widely used ontology languages such as OIL, DAML+OIL and OWL (Horrocks, 2003). The key motivation for basing ontology

[20] http://kaon.semanticweb.org

languages on DLs is that DL systems can then be used to provide computational services for ontology tools and applications (Knublauch *et al.*, 2004a). The increasing use of ontologies, along with increases in their size and complexity, brings with it a need for efficient DL inference engines. Given the high worst case complexity of the satisfiability/subsumption problem for the DLs in question, optimizations that exploit the structure of typical ontologies are crucial to the viability of such reasoners (Horrocks *et al.*, 2004).

Inference is the act or process of deriving a logical consequence conclusion from premises. That is, deriving facts that are not expressed in ontology or in knowledge base explicitly. Inference in ontologies and knowledge bases are normally supported by inference engines also named as reasoners.

Ontologies as knowledge modeling acts based on organization of concepts:

> *Identification:* the ability to recognize e.g., an object or an action, as belonging to a category;
>
> *Specialization* and *generalization*: the ability to memorize abstractions of categories differentiated in hierarchies of specialization/generalization e.g., "nature origin objects, apple, orange, strawberries are fruit", "technological products with fruit name, are electronic objects". These hierarchies are the basis of inferences at the heart of information retrieval and exchange, e.g., "orange is a fruit", "iphone is an electronic device".

Thus, this structure in hierarchical categories with related identification and specialization/generalization is used to capture a consensus which is socially and culturally dependent. This background knowledge is lacking in information systems relying only on terms and plain-text search. A possible approach is thus to make this knowledge explicit and capture it in logical structures that can be exploited b automated systems. This is main purpose of an ontology (Friedman-Hill and Scuse, 2008): to capture the semantics and relations of the concepts that humans use, make them explicit and eventually code them in symbolic systems so that they can be manipulated and exchanged.

Ontologies inference capability is one of the reasons why a specification needs to be formal (Sharma and Osei-Bryson, 2008).

Inference may be inductive or deductive (Chu and Hwang, 2008)(Bombardier *et al.*, 2007):

- *Inductive*: the process by which a conclusion is inferred from multiple observations is called inductive reasoning. The conclusion may be correct or incorrect, or correct, or correct to within certain degree of accuracy, or correct in certain situations. Conclusions inferred from multiple observations may be tested by additional observations.
- *Deductive*: The process by which a conclusion is logically inferred from certain premises is called deductive reasoning. Mathematics makes use of deductive inference. Certain definitions and axioms are taken as a starting point, and from these certain theorems are deduced using pure reasoning. The idea for a theorem may have many sources: analogy, pattern recognition, and experiment are examples of where the inspiration for a theorem comes from. However, a conjecture is not granted the status of theorem until it has a deductive proof. This method of inference is even more accurate than the scientific method. Mistakes are usually quickly detected by other mathematicians and corrected. The proofs of Euclid, for example, have mistakes in them that have been caught and corrected, but the theorems of Euclid, all of them without exception, have stood the test of time for more than two thousand years.

Artificial Intelligence systems first provided automated logical inference and these were once extremely popular research topics, leading to industrial applications under the form of expert systems and later business rule engines (Sharma and Osei-Bryson, 2008). An inference system's job is to extend a knowledge base automatically (Horrocks *et al.*, 2005). The knowledge base is a set of propositions that represent what the system knows about the world (Kishore *et al.*, 2004). Several techniques can be used by that system to extend knowledge base by means of valid inferences (Tsarkov and Horrocks, 2004). An additional requirement is that the conclusions the system arrives at, are relevant to its task.

The main job of an inference engine is to check whether a certain proposition can be inferred from a knowledge base using an algorithm called backward chaining (Parsia and Sirin, 2004). Let us return to our Socrates syllogism. We enter into our Knowledge Base the following piece of code:

mortal(X) :- man(X).man(socrates).

Here *:-* can be read as *if*. Generally, if P→Q (if P then Q) then correspondent code *Q :- P (Q if P)*. This states that all *men* are mortal and that *Socrates* is a *man*. Now we can ask the reasoner system about *Socrates*:

?- mortal (socrates).

where *?-* signifies a query: *Can mortal(socrates)?*.
Be deduced from the knowledge base using the rules gives the answer "Yes". On the other hand, asking the reasoner system the following:

?- mortal(plato)

gives the answer "No". This is because reasoner does not know anything about Plato, and hence defaults to any property about Plato being false (the so called closed world assumption).
Finally *?- mortal(X)* (*Is anything mortal?*) would result in "Yes" (and in some implementations: "Yes"*:X=socrates*) reasoner can be used for vastly more complicated inference tasks.

Automatic inference and the semantic web
Recently automatic reasoners found in semantic web a new field of application. Being based upon first-order logic, knowledge expressed using one variant of OWL can be logically processed, that is, inference can be made upon it.

Nevertheless inference engines are created with the focus on tractable reasoning. A few examples of tasks required from inference engines are as follows:

Satisfiability of a concept - determine whether a description of the concept is not contradictory, i.e., whether an individual can exist that would be instance of the concept;

Subsumption of concepts - determine whether concept *C* subsumes concept *D*, that is, whether description of *C* is more general than the description of *D*;

Consistency of axioms - determine whether individuals do not violate descriptions, relations and restrictions.

Check an individual - check whether the individual is an instance of a concept

Individuals retrieval - find all individuals that are instances of a concept;

Realization of an individual - find all concepts which the individual belongs to, especially the most specific ones.

However these tasks are not semantically very different, e.g., *satisfiability* can be tested as *subsumption* of *bottom - concept* is *unsatisfiable* if no individual can exist that would be instance of the concept. For all tasks, it is enough to be able to check deductive consequence or derive all deductive consequences of a theory. However, there may be special optimized algorithms for different tasks in a reasoner.

2.3.8 Inference engines - reasoners

At this section we present a survey on different kinds of existing inference engine implementations that can be used for reasoning within the different ontologies. We survey description logic reasoners from the areas of logic programming.

Description Logic Reasoners

The basic inference problems in description logics reasoning are (Predoiu and Grimm, 2006):

- *Knowledge Base Consistency* - a knowledge base is consistent if it has a model, i.e. it can be interpreted in the model-theoretic sense. A knowledge base that contains a contradiction does not have a model;

- *Concept Satisfiability*: a concept is satisfiable with respect to a knowledge base if this knowledge base can be interpreted such that the extension of the concept is non-empty, i.e. the concept can potentially have an instance;
- *Concept Subsumption:* a concept A subsumes a concept B with respect to a knowledge base if any instance of B is also an instance of A, no matter how the knowledge base is interpreted.
- *Concept Equivalence*: A concept A is equivalent to a concept B with respect to a knowledge base if A and B subsume each other.
- *Concept Disjointness:* two concepts are disjoint with respect to a knowledge base if they don't have a common instance, no matter how the knowledge base is interpreted.

All these inference tasks can be reduced to the main inference of checking a knowledge base for consistency, which can be realized by checking the set of facts in the knowledge base for unsatisfiability (Horrocks, 2003). All state-of-the-art reasoners for DL implement the tableau mechanism for performing this check (Predoiu and Grimm, 2006). The basic idea of the tableau method is to construct a model-theoretic interpretation for the set of facts to be checked according to the constraints these facts impose on the individuals in this interpretation, which is either successful or leads to a contradictory situation. In case of success, the thus constructed interpretation is a model for the set of facts in the knowledge base. All the following DL reasoners implement the tableau method for realizing all the basic inference tasks listed above.

Pellet Reasoner

Pellet is an OWL DL reasoner based on the tableaux algorithms developed for expressive Description Logics (Parsia and Sirin, 2004). It supports the full expressivity OWL DL including reasoning about nominal's (enumerated classes)

Pellet supports reasoning with the full expressivity of OWL-DL, and has been extended to support the forthcoming OWL 2 specification, which adds the following language constructs: qualified cardinality restrictions; complex sub-property axioms (between a property chain and a property); local reflexivity restrictions; reflexive, "irreflexive",

symmetric, and anti-symmetric properties; disjoint properties; negative property assertions; vocabulary sharing (punning) between individuals, classes, and properties; user-defined data-ranges

Pellet also provides reasoning with the following features from OWL Full: the inverse functional datatype properties.

All the standard inference services that are traditionally provided by DL reasoners, are also available in Pellet:

Consistency checking: Ensures that an ontology does not contain any contradictory facts;

Concept satisfiability: determines whether it's possible for a class to have any instances. If a class is unsatisfiable, then defining an instance of that class will cause the whole ontology to be inconsistent;

Classification: computes the subclass relations between every named class to create the complete class hierarchy. The class hierarchy can be used to answer queries such as getting all or only the direct subclasses of a class;

Realization: finds the most specific classes that an individual belongs to; i.e., realization computes the direct types for each of the individuals. Realization can only be performed after classification since direct types are defined with respect to a class hierarchy. Using the classification hierarchy, it is also possible to get all the types for each individual.

FACT Reasoner

The FaCT (Fast Classification of Terminologies) reasoner system is a Description Logic (DL) classifier that can also be used for modal logic satisfiability testing (Tsarkov and Horrocks, 2006) (Udrea *et al.*, 2007). The FaCT system includes two reasoners, one for the logic SHF (ALC augmented with transitive roles, functional roles and a role hierarchy) and the other for the logic SHIQ (SHF augmented with inverse roles and qualified number restrictions), both of which use sound and complete tableaux algorithms.

FaCT++ is a DL reasoner designed as a platform for experimenting with new tableaux algorithms and optimization techniques (Tsarkov and Horrocks, 2004). It incorporates standard optimization techniques and also a "ToDo list" architecture that is better suited to more complex tableaux algorithms (such as those used to reason with OWL ontologies), and allows for a wider range of heuristic optimizations (Tsarkov and Horrocks, 2006)(Hunyadi and Pah, 2008).

FaCT's most interesting features are (Horrocks *et al.*, 2005) (Tsarkov and Horrocks, 2006):

- its expressive logic (in particular the SHIQ reasoner): SHIQ is sufficiently expressive to be used as a reasoner for the DLR logic, and hence to reason with database schemata;
- its support for reasoning with arbitrary knowledge bases (i.e., those containing general concept inclusion axioms);
- its optimized tableaux implementation (which has now become the standard for DL systems), and its CORBA based client-server architecture.

2.4 Related work

KDD is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad *et al.*, 1996). In a naïf discourse mode we may say that, KDD is a process which includes several steps, most of which can be realized by automatic data intensive computations. However, as a nontrivial process, human capabilities and judgment is still a fundamental ingredient to ensure that useful and valid knowledge is derived from the data. Nowadays, human capabilities assume the form of skills and expertise in different domains such as databases, statistics, machine learning, data mining, as well as the specific business/application domain.

Thus, in order to manage a knowledge discovery project, an ontology focusing all related KDD processes knowledge is worth being constituted. Such ontology will be used to assist the KDD process and therefore, to ensure an accomplishment of related tasks reducing, or at least, controlling expenditures for KDD projects.

KDD domain is a special type of domain (Diamantini *et al.*, 2006a). Consequently, we refer to the KDD ontology as a special type of ontology. KDD ontology is a conceptualization of the KDD domain in terms of tasks, techniques, algorithms, tools and tool properties (such as performance) and the kind of data that can be used for (Cannataro and Comito, 2003) (Nigro *et al.*, 2008). As such, KDD ontology has a similar role according to any business domain ontology: it helps the business expert to understand the KDD domain, so that he can either effectively collaborate with a KDD expert in the design of a KDD project, or design the KDD project on his own. In this case, the KDD ontology can support the user in browsing a tool repository that is organized with according it.

In order to face a KDD project, expertise on both the application world and the KDD world is needed. Hence, when talking about domain knowledge, we mean knowledge for the application (business) domain as well as for the KDD domain. Application

domain holds information about all the objects involved in the application (Diamantini *et al.*, 2006b). In addition, such a domain possesses knowledge about connections among objects, constraints and hierarchy of objects, and should describe goals and activities to be performed on objects in order to achieve stated goals, e.g., in a DBM project, objects may include: raw data, detailed personal customer information or technical product data - raw data is simultaneously linked to (kept in) stores or providers and (exploited in) customers or sales.

KDD ontology involves several issues (Diamantini *et al.*, 2004) (Diamantini *et al.*, 2006a): manage different data sources; integrate information and knowledge produced during the KDD project; orchestrate different tools; move efficiently the huge amount of KDD data to analyze; among others.

The use of domain ontologies is proposed to guide the KDD process and to give support to domain experts (Smith and Welty, 2001) (Phillips and Buchanan, 2001). Such ontology would be able to support the extraction of novel features, by exploiting relations among domain concepts. The use of ontologies in KDD field is normally proposed to refine the induced knowledge and to correctly interpret the results (CeSpivova *et al.*, 2004) (Cellini *et al.*, 2007) (Brezany *et al.*, 2008). Others authors propose specific KDD task oriented ontologies to support specific work, like algorithms selection, data or data quality tasks selection.

A KDD assistance through ontologies should provide user with nontrivial, personalized "catalogs" of valid KDD-processes, tailored to their task at hand, and helps them to choose among these processes in order to analyze their data. In literature there are relevant contributions focusing the integration of ontologies and the KDD process, arguing they importance (Vilalta *et al.*, 2005) (Brazdil *et al.*, 2009). Nevertheless mostly of such contributions are focused on "the role of domain knowledge in KDD" or centering their focus at DM level, regarding topics like classification algorithms performance measure or algorithms optimization.

Ontologies have recently emerged playing an important role in the knowledge engineering research from different research areas:

- The integration of ontologies and KDD is suggested as the most promising approach for constrains knowledge discovery and for avoiding the well-known problem of data over fitting by the discovered models (Domingos, 2003);
- To identify and simplify the KDD process tasks, like, attributes description relationship rules, hierarchical generalization trees and constrains (e.g., the specification of degrees of confidence in the different sources of evidence) (Anand *et al.*, 2007);
- To assist the process of Data Mining (DM) through a systematic enumeration of valid DM processes and with an effective ranking of valid processes by different criteria, to facilitate the choice of DM processes to execute (Bernstein *et al.*, 2005);
- Ontologies are developed according with the collection, presentation and use of knowledge. They include various concepts, facts, data, graphs and other information forms, related to the domain (Kasabov *et al.*, 2007);
- Ontologies may be used as a guide to gradually accumulate knowledge in order to construct a domain knowledge base in the iterative process of a KDD task (Phillips and Buchanan, 2001);
- An ontology for the DM domain can be used to simplify the development of distributed knowledge discovery applications , offering a domain expert a reference model for the different type of data mining tasks, methodologies and software available to solve a given problem, helping a user in finding the most appropriate solution (Cannataro and Comito, 2003) (Brezany *et al.*, 2008).

In spite of the increase of investigation in the integration of domain knowledge, by means of ontologies and KDD, most approaches focus mainly in the DM phase of the KDD process while the role of ontologies in other phases of the KDD has been relegated. Currently, there are other approaches being investigated in the ontology and KDD integration, like ONTO4KDD and KDD4ONTO[21] or AXIS[22]. Both of them are

[21] http://olp.dfki.de/pkdd04/cfp.htm

focusing the application of ontologies in order to improve overall KDD process regarding DM models optimization and sophistication. In the literature there are several knowledge discovery life cycles, mostly reflect the background of their proponent's community, such as database, artificial intelligence, decision support, or information systems (Gomez-Perez *et al.*, 2004). For this we´ve adopted the KDD framework (Fayyad *et al.*, 1996) and based our domain knowledge on the results of the Delphi method (Rowe and Wright, 2001) (Chu and Hwang, 2008) developed with DBM specialists.

Although the scientific community is addressing ontologies and KDD improvement, at the best of our knowledge, there isn't at the moment any fully successful integration of these.

This research encompasses an overall perspective, from business to knowledge acquisition and evaluation. To do this we use the Data Mining Ontology (DMO), integrated in the KDD process to propose a general framework. Moreover, this research focuses the KDD process regarding the best fitting modeling strategy selection supported by ontology.

Therefore, in this work we focus the role of ontology in order to assist the KDD in different stages of the process: data understanding; data preparation and modeling. Indeed, to select the appropriate and adequate task sequence to support the KDD work becomes an important decision. This work proposes a computational model based on ontologies to assist the KDD planning process.

[22] http://ralyx.inria.fr/2006/Raweb/axis/uid4.html

III

Research Approach

Along this chapter we present the general research guidelines that we have followed in order to achieve the research objectives. Firstly, we briefly introduce the main developed research focus and then, each main research focus is detailed in order to provide a full understanding the developed work.

3.1 Approach

This research holds three different scientific areas: database marketing, knowledge discovery in databases (KDD) and ontologies. Ontologies are the core of the research serving as an integrator element from the domain research area knowledge (DBM) and main techniques resources (KDD).

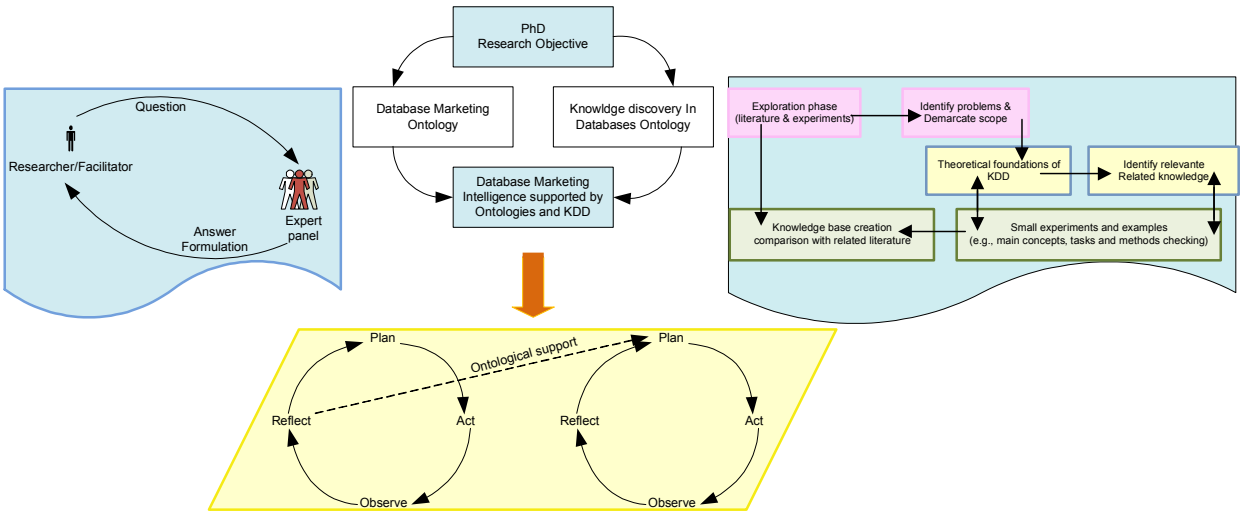


Figure 8: General developed work framework

Attaining the aforementioned main research objective we have programmed our work into two main parts (Figure 8):

Ontology development - regarding all tasks required to create both ontologies: DBM ontology and KDD ontology; and

System prototype - focusing the ontology integration and use in DBM projects in order to effectively support the KDD process.

Concerning the former approach, we have used two different methodological strategies focusing each ontology development:

Database marketing ontology - Delphi Method & 101 methodology for ontology development. Delphi methodology to modulate all DBM related knowledge. Firstly, we focused our work on DBM knowledge, in order to structure its main concepts and systematize the overall organization, namely, relationship marketing objectives and activities and main marketing database data types. Such research work has provided enough information to effectively respond to the first 101 methodology items. Then, following the 101 ontology development methodology, we proposed a DBM ontology (DBMO);

Knowledge Discovery in Databases Ontology - Methontology and literature review work in order to achieve a full KDD process understanding and ontology formalization. Here, we have explored, as much as possible all scientific works published in order to identify data oriented tasks, Data Mining (DM) methods and algorithms, evaluation metrics and strategies. Methontology framework has been used to incorporate all operational and background KDD knowledge ending with the KDD ontology formalization.

Regarding the system's prototype development we have conducted our research through Action-Research methodology (Figure 8). To prototype the Database Marketing Intelligence (DBMI), we had focused the ontologies integration for both levels: DBM process and KDD process support

3.2 Ontologies development

Noticeably, there is no one correct way or methodology to develop ontologies. For this we have studied different development methodologies. From this research, two main groups could be identified. On the one hand, there are experience-based methodologies, such as the methodology based on TOVE (Toronto Virtual Enterprise) (Fox and Gruninger, 1997) or on Enterprise Model (Uschold and King, 1995). On the other hand, there are methodologies prescribing dynamic prototypes models, such as METHONTOLOGY (Fox and Gruninger, 1997) (Fernandez *et al.*, 1997) (Gomez-Perez *et al.*, 2004) that proposes a set of activities to develop ontologies based on its life cycle and the prototype refinement; or, the 101 Method (Jones *et al.*, 1998) (Noy and McGuinness, 2003) that proposes an iterative approach to ontology development. Usually, the first ones are more appropriate when purposes and requirements of the ontology are clear, the second one is more useful when the environment is dynamic and difficult to understand – attaining our research objective which focuses the ontology used within the practical process research, it is the most appropriate. Moreover, it is common to merge different methodologies with other approaches since each one of them provide design ideas that distinguish it from the rest. Also, this merging work depends on the ontology users and ontology goals.

This research aims to propose a methodology based on ontologies and the knowledge extraction process. Indeed, we attain to develop a methodology based on KDD and DBM ontological assistance. Therefore, we have organized the research program according it: firstly, we have developed the DBM ontology, then we have proceeded to the development of the KDD ontology for knowledge extraction assistance and finally we have integrated both into a new DBM intelligence methodology.

Regarding ontologies development, we have taken different approaches for the different developed ontologies:

- we have used the Delphi method and the 101 methodology for the database marketing ontology formulation. Due to the absence of related work in the DBM field, we decided to apply the Delphi method in order to get the required domain

knowledge and, therefore, we have used the 101 development method (Noy and McGuinness, 2003);

- For the KDD ontology we have based on the literature review and methontology method. In contrast to the previous, we have found a much documented research area allowing us to construct our general KDD knowledge base from the published scientific literature (theoretical and practical published works). Then we have procedded with the ontology development through the methontology methodology.

A catalog of criteria, common to both ontologies was considered in their development. Such criteria has been proposed and analyzed (e.g., (Jurisica *et al.*, 1999) or (Gruber, 1993)) and they are briefly outlined in the following list (Kalfoglou and Robertson, 2000, Kalfoglou and Schorlemmer, 2007):

Clarity, refers to the effective communication of the intended meaning. Formalism has been proposed as a means to dissipate ambiguities, i.e., whenever possible, a definition can be stated as a logical axiom. However, all definitions should be documented in natural language.

Coherence, means that the ontology should endorse all the inferences that are consistent with the axioms. Not only should the defining axioms be logically consistent, but the concepts should also be defined informally (such as documentation and examples). If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

Extendibility, the ontology should be designed to anticipate the shared uses of its vocabulary. One should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.

Minimal encoding bias, an encoding bias arises when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized because the ontology can be shared by agents or systems using different representation schemes and different implementation languages.

Minimal ontological commitment, an ontology should make as few claims as possible about the world being modeled, allowing the parties using the ontology freedom to specialize and instantiate it as needed.

It has to be noted that ontology designers cannot always comply with the above criteria. A number of tradeoffs can be necessary (Nigro *et al.*, 2008) and ways of compromising between well designed ontologies and applicability have been investigated (Borst *et al.*, 1997).

Ontology development is necessarily an iterative process. Moreover, there is no one "correct" way or methodology for developing ontologies (there are always viable alternatives - the best solution almost always depends on the application objective) (Noy and McGuinness, 2003). However, in general, their development can be divided into two main phases (Guarino, 1995) (Jasper and Uschold, 1999): *specification* and *conceptualization*. The goal of the specification phase is to acquire domain knowledge. The goal of the conceptualization phase is to organize and to structure this knowledge using external representations that are independent of the implementation languages and environments.

In order to define the KDD ontology initially we carried out an extensive literature review work regarding the specification and used the analysis steps from Methontology in the conceptualization process. In this work, both approaches were merged because on one hand, core requirements are clear but on the other, domain complexity drives to adopt an iterative approach to manage refinement and extensibility. Moreover, we have considered an incremental construction that allows to a more refining original model in successive steps and they offer different representations for the conceptualization task.

To develop the DBM ontology we have used the Delphi method (Delbecq *et al.*, 1975)(Cochran, 1983)(Linstone and Turoff, 2002)(Bonnemaizon *et al.*, 2007)(Chu and Hwang, 2008) to acquire domain knowledge and to structure tasks regarding the complete specification. In order to carry out the conceptualization work we have used the 101 method (Jones *et al.*, 1998) (Noy and McGuinness, 2003). Once we achieved our first

ontology we have submitted it again to the expert panel for approval. Thereafter some refinement work we achieved a consensual KDD ontology.

3.2.1 Delphi methodology

The Delphi method is normally used to structure a group communication process to deal with and to build consensus about a particular and complex topic. The method works based on an expert panel group (anonymous experts - no expert knows who is else is on the panel) who answer to proposed questions and formulate a set of hypotheses about it (Figure 9). Then, the method is developed based on a dialectical inquiry approach (Armstrong, 2006): the researcher introduces a set of questions in order to establish an opinion or point of view from the expert panel. Then the expert panel (individually) answers reporting a formulation (conflicting opinion or point of view). The researcher in charge has to generate a synthesis (a new agreement or consensus) and submit it again to the expert panel. This loop only ends when the researcher achieves a consensus with all expert panel members (Rowe and Wright, 2001).

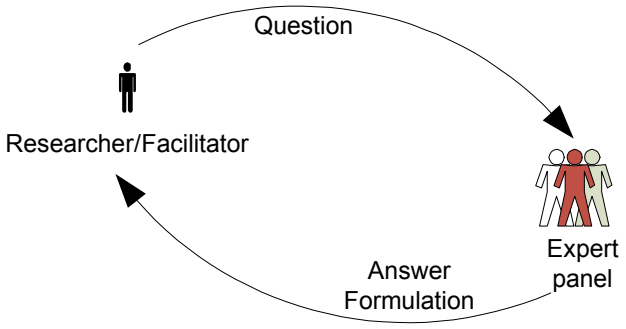


Figure 9: Delphi methodology process

In order to constitute our expert panel we have addressed many invitations to community groups (e.g., Portuguese marketing professors association or Portuguese professional marketing association) and to individuals who have recognized experience within DBM field.

At the end of this initial process we achieved the agreement of 7 different personalities of different professional and academic skills: two professionals from petroleum distribution, one other professional from cable television and one other expert from a marketing agency. This panel also counts with experts from marketing research and one PhD relationship marketing student.

Currently, Delphi is considered a useful method for eliciting and aggregating expert opinion whenever there is a lack of viable or practical statistical techniques (Cochran, 1983)(Murry and , 1995)(Bonnemaizon *et al.*, 2007). It can be defined as a mid-term qualitative forecasting method that is based on building a consensus amongst a group of experts (Armstrong, 2006). A Delphi type study enables an exchange of information amongst experts over a number of rounds (iterations) and allows experts to react to the information gathered during each round and to fine-tune their forecast by means of a feedback mechanism (controlled retroaction). Beyond these three main principles (anonymity – iteration – retroaction), the method's validity is firstly based on a rigorous selection of experts whose combined knowledge and expertise must reflect the full scope of the problem area.

Some authors have suggested asking the persons involved to estimate their own degree of expertise, with others considering that the level of expertise does not necessarily need to be high (Rowe and Wright, 2001). Delphi's validity is also dependent on the size of the group of experts (Vernette, 1997) (research suggests, that the minimum threshold is 5–7 experts, and that a range of 8–10 offers the best precision/cost ratio). Besides experts, information contributions are marginal (Zairate *et al.*, 2006). The method's validity relies on a strict implementation of the process: three iterations are usually needed to obtain a satisfactory consensus (Armstrong, 2006).

3.2.2 101 Methodology

101 Methodology is based on the principle that, there are several viable alternatives for ontology development, and ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.

Ontologies have become core components of many large applications. This methodology presents a set of tasks for creating ontologies based on declarative knowledge representation. It leverages the author's experiences developing and maintaining ontologies in a number of ontology environments including Protégé [23], or Ontolingua[24]. The Ontology 101 methodology is relatively simple, since it defines simple and, some of them, generic steps. Indeed this methodological approach does not make assumptions about knowledge representation/ontology language.

This methodology uses the ontology domain and scope (based on related knowledge) to pragmatically determine which the best approach for ontology development is. Also, this methodology assumes that the ontology development is a process of iterative design that will likely continue through the entire lifecycle of the ontology.

101 methodology uses a generic approach from the domain application to the effective instance creation, through the following steps:

Determining the domain and scope of the ontology;
Considering reusing existing ontologies;
Enumerating important terms in the ontology;
Defining the classes and the class hierarchy;
Defining the properties of classes;
Defining the facets of the slots;
Creating instances;

[23] http://protege.stanford.edu/
[24] http://www.ksl.stanford.edu/software/ontolingua/

3.2.3 Literature review research based method

Knowledge Discovery in Databases (KDD) is accepted among computer scientists as a process that allows to select, explore and extract valid and useful information from databases. Since this research area is very well documented through scientific books, journals or proceedings, we have supported our KDD ontology construction on those published works.

Terms in ontologies are selected with great care, ensuring that the most basic (abstract) foundational concepts and distinctions are defined and specified. The terms chosen form a complete taxonomic set. The relationships among terms are defined using formal techniques. These formally defined relationships provide the semantic basis for the terminology selection.

Although taxonomy contributes to the semantics of a term in a vocabulary, ontologies include richer relationships between terms. These rich relationships enable the expression of domain-specific knowledge, without the need to include domain-specific terms.

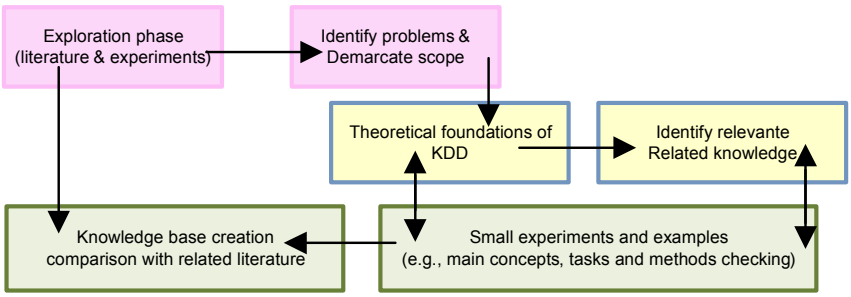


Figure 10: literature review research based: method used

To do this we carried out an exhaustive literature review research (Figure 10) in order to get the aforementioned ontology requirements (more in ontologies background chapter).

The work developed started with an exhaustive literature review regarding theoretical and experimental contributions, providing us some directions to the problems and scope

identification. Then, we proceeded with KDD theoretical terms selection in order to enumerate relevant KDD related knowledge. In each step of this work we have made some experiments, demanding to evaluate their pertinence and subsequent universality (common term understanding and acceptance).

Hence, our KDD knowledge base creation is a result of the identification work of relevant concepts and relationships between concepts through literature.

3.2.4 Methontology Methodology

This methodology was developed within the Ontology group at Universidad Politécnica de Madrid. Methontology (Fernandez *et al.*, 1997)(Blazquez *et al.*, 1998)(Gomez-Perez *et al.*, 2004) enables the construction of ontologies at the knowledge level. It has its foundations in the main activities identification in the software development process and in knowledge engineering methodologies (Gomez-Perez and Rojas-Amaya, 1999).

The building ontology's process may span some problems issues, like: problem specification; domain knowledge acquisition and analysis; conceptual design and commitment to community ontologies; iterative construction and testing; publishing the ontology as a terminology; and, possibly populating a conforming knowledge base with ontology individuals. Although the process may strictly be a manual exercise, there are tools available that can automate portions of it.

Ontology development cycle is proposed as a stable approach to conceive and understand the process that aims to produce an ontology. The usually accepted main phases through which an ontology is built are knowledge acquisition, evaluation and documentation (Blazquez *et al.*, 1998, Corcho *et al.*, 2003, Fernandez *et al.*, 1997, Gomez-Perez *et al.*, 2004, Lopez, 1999, Lopez *et al.*, 1999):

- *Knowledge acquisition:* refers to the acquisition of knowledge about the subject either by using elicitation techniques on domain experts or by referring to relevant bibliography. Several techniques can be used to acquire knowledge, such

as brainstorming, interviews, questionnaires, text analysis, and inductive techniques;

- *Evaluation:* relate all activities concerning with the ontology operation. Technically judge the quality of the ontology;
- *Documentation:* register and report what was done, how it was done and why it was done. Documentation associated with the terms represented in the ontology is particularly important, not only to improve its clarity, but also to facilitate maintenance, use and reuse

3.2.5 System Prototype Design

We have adopted an action research methodology approach for the system prototype design and development. Action research is a self-reflective, self critical and critical enquiry undertaken by professionals to improve the rationality and justice of their own practices, their understanding of these practices and the wider contexts of practice (Lomax, 2002). Moreover, action research methodology contributes to the development and improvement of systems. This methodology incorporates the four-step process (Figure 11) of planning, acting, observing and reflecting on results from a particular project or body of work (Zubber-Skerritt, 2000), (O'Brien, 2002). The concept essentially concerns with a group of people who work together to improve their work processes (Baskerville, 1999), (Carson *et al.*, 2004).

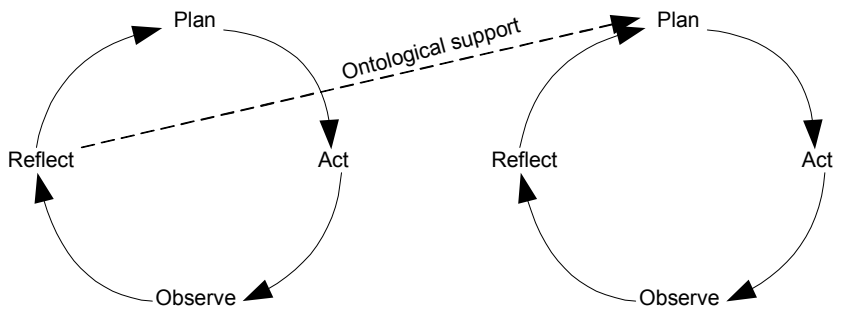


Figure 11: Action research methodology

This choice of action research was based on two counts. First, due to the low number of scientific research works that has been conducted on supporting KDD process over intelligent structures like ontologies. Second, ontologies can play an important role in the knowledge development as long as they register past knowledge for future reuse (Figure 11). Thus exploratory research was required and action research provides this capability better than many other alternatives (Dick, 2008).

Nevertheless, first we need to formulate, test, deploy and evaluate a complete DBM project whereas DBM is developed over a KDD process. Hence, we focus such interaction and annotate it in a semantic language, like RDF (Resource Description Framework) or OWL (Ontology Web Language) and use the SWRL (Semantic Web Language Rule) to infer.

Due to its ontological characteristic, this stage of the project turned out to be emancipator action research (Leary *et al.*, 2004) rather than merely technical or practical (Zuber-Skerrit and Perry, 2000). The relationship between the research group elements (namely, marketers from participant organization) was collaborative.

Action research methodology holds some strategies to validate results. Some of them are as follows (Merriam, 1998): triangulation, member checks, long-term observation, peer examination, and participatory or collaborative modes of research, researcher's biases clarified at outset.

This action research work attains to develop a DBMI prototype system. That could effectively improve the DBM process, supported by ontologies and KDD, e.g., to transform some business models, like ineffectual relationships to a more successful client relationship, based on individual profile.

Besides, due to its interactive nature, action research, may contribute to the system development and also to the improvement of some marketing objectives like: differentiating, interacting, personalizing and also learning from each interaction between customer and organization.

V

Developed Work and Contribution

This chapter introduces all developed work and presents the achieved contribution. The research work is introduced in the form of some original articles published in journals, chapters of books and proceedings of international conferences.

The developed work was organized in terms of key operational blocks. This approach meets the research methodologies' directives (chapter III) allowing the publication and, consequently, the discussion of self-contained research parts. Each part and correspondent contributions have been subject to validation by the international research community in conferences and journals where the works were published. The parts considered were:

- Ontology development: work was developed towards the formulation of two ontologies: firstly the KDD ontology formulation, which uses thorough and exhaustive literature review to abtain all related methods, tasks and approaches for the KDD development; and secondly, the DBM process supported by KDD ontology formulation, whereas the former KDD ontology was reused and integrated. To this end, we have been present at Stanford University Protégé-OWL research centre (United States) , in order to improve and test our work;
- KDD development and ontologies use. Some practical work was performed towards the effective KDD assistance in a DBM project context. We have been in the Database Marketing Research Centre of Ghent University (Belgium), in order to carry out some test with marketing databases. We have developed some KDD supported DBM processes and assisted by ontologies, in order to test and

evaluate the extent of ontologies to effectively support and assist data analyst in each KDD process phase. ;

- Methodology development. Since we have achieved our major research findings, we have proceeded to the design and construction of the methodology. To do this we have systematized our findings in terms of multi-layer based system.

The work developed will be presented as a list of published papers related to the above identified parts:

- An Ontology Proposal for Knowledge Discovery in Databases (section 4.1.4);
- Ontology Supported Database Marketing (section 4.2.4);
- Ontological Assistance for Knowledge Discovery Databases Process (section 4.3.4);
- Database Marketing Intelligence Supported by Ontologies (section 4.4.4).

Preceding each research work, there is an introduction preamble that intends to explain the framework of each work, in order to complement the description of methodologies, resources and methods, and also to resume some of the most relevant results.

4.1 Knowledge Discovery in Databases Ontology

4.1.1 Introduction

In this section we present the developed work towards knowledge discovery in databases ontology creation. In spite of our efforts such ontology focuses a general approach to a complete KDD ontology. Therefore, we focus the main general KDD process.

KDD is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad *et al.*, 1996). In a naïf discourse mode we may say that KDD is a process which includes several steps, most of which can be realized by automatic data intensive computations. However, as a nontrivial process, human capabilities and judgment is still a fundamental ingredient to ensure that useful and valid knowledge is derived from the data. Nowadays, human capabilities assume the form of skills and expertise in different domains such as databases, statistics, machine learning, data mining, as well as the specific business/application domain.

Thus, in order to manage a knowledge discovery project, an ontology focusing all related KDD process knowledge is worth being constituted. Such an ontology will be used to assist the KDD process and therefore to ensure the accomplishment of related tasks reducing, or at least, controlling expenditures for KDD projects.

4.1.2 Research approach

KDD domain is a special kind of domain (Diamantini *et al.*, 2006a), consequently, we refer to the KDD ontology as a special kind of ontology. KDD ontology is a conceptualization of the KDD domain (Euler and Scholz, 2004)(Diamantini *et al.*, 2004)(Nigro *et al.*, 2008) in terms of tasks, techniques, algorithms, tools and tool properties (such as performance) and the kind of data that it can be used for (Fisher, 1987)(Kotasek and Zendulka, 2000)(Cannataro and Comito, 2003) . As such, KDD ontology has a similar role regarding any business domain ontology (Kuo *et al.*, 2007b)(Phillips and

Buchanan, 2001): it helps the business expert to understand the KDD domain, so that he can either effectively collaborate with a KDD expert in the design of a KDD project, or design the KDD project on his own. In this case, the KDD ontology can support the user in browsing a tool repository that is organized according to it.

In order to face a KDD project, expertise in both the application world and the KDD world is needed. Hence, when talking about domain knowledge, we mean knowledge for the application (business) domain as well as for the KDD domain. Application domain holds information about all the objects involved in the application (Diamantini *et al.*, 2006b). In addition, such a domain possesses knowledge about connections among objects, constraints and hierarchy of objects, and should describe goals and activities to be performed on objects in order to achieve stated goals, e.g., in a DBM project, objects may include: raw data, detailed personal customer information or technical product data - raw data is simultaneously linked to (kept in) stores or providers and (exploited in) customers or sales.

KDD ontology involves several issues (Diamantini *et al.*, 2004) (Diamantini *et al.*, 2006a): managing different data sources; integrating information and knowledge produced during the KDD project; orchestrating different tools; efficiently moving the huge amount of KDD data to analyze; among others.

Since our work focuses on the integration of knowledge discovery techniques and ontologies at database marketing process, we have developed an ontology in order to effectivly support the KDD process. We have carried out a double approach method development. Initially we performed an exhaustive literature review work. Then, at a second step we have used the methontology methodology in order to develop the KDD ontology.

Literature review research based method

KDD is accepted among computer scientists as a process that allows selecting, exploring and extracting valid and useful information from databases. Bearing in mind this research area is very well documented through scientific books, journals or proceedings, we have supported our KDD ontology construction on those published works.

Terms in ontologies are selected with great care, ensuring that the most basic (abstract) foundational concepts and distinctions are defined and specified. The terms chosen form a complete taxonomic set and the relationships among terms are defined using formal techniques. It is these formally defined relationships that provide the semantic basis for the terminology chosen.

Although taxonomy contributes to the semantics of a term in a vocabulary, ontologies include richer relationships between terms. These rich relationships enable the expression of domain-specific knowledge, without the need to include domain-specific terms. To achieve this we have carried out an exhaustive literature review research in order to get the aforementioned ontology requirements.

We have started by the exploring all literature regarding theoretical and experimental contributions, providing to us some directions to the problems and scope identification. Then, we proceeded with KDD theoretical terms selection in order to enumerate relevant KDD related knowledge. At each step of this work we have made some experiments, aiming to evaluate their pertinence and subsequent universality (common term understanding and acceptance).

Our knowledge base creation is therefore a result from an identification work of relevant concepts and relationships between concepts through literature.

Methontology Methodology

Methontology methodology (Fernandez *et al.*, 1997)(Blazquez *et al.*, 1998)(Gomez-Perez *et al.*, 2004) enables the construction of ontologies at the knowledge level. It has its roots in the main activities identified by the software development process and in knowledge engineering methodologies (Gomez-Perez and Rojas-Amaya, 1999).

The building of an ontology's process may span problem specification, domain knowledge acquisition and analysis, conceptual design and commitment to community ontologies, iterative construction and testing, publishing the ontology as a terminology, and possibly populating a conforming knowledge base with ontology individuals. While the process may strictly be a manual exercise, there are tools available that can automate portions of it.

Ontology development cycle is the most time stable approach to conceive and understand the process that aims to produce an ontology. The usually accepted main phases through which an ontology is built are knowledge acquisition, evaluation and documentation:

- *Knowledge acquisition:* refers to the acquired knowledge about the subject either by using elicitation techniques on domain experts or by referring to relevant bibliography;
- *Evaluation:* relate all activities concerning with ontology operation. Technically judge the quality of the ontology;
- *Documentation:* register and report of what was done, how it was done and why it was done.

4.1.3 Results

As results we have achieved an explicit KDD ontology which integrates background and practical knowledge (Figure 12).

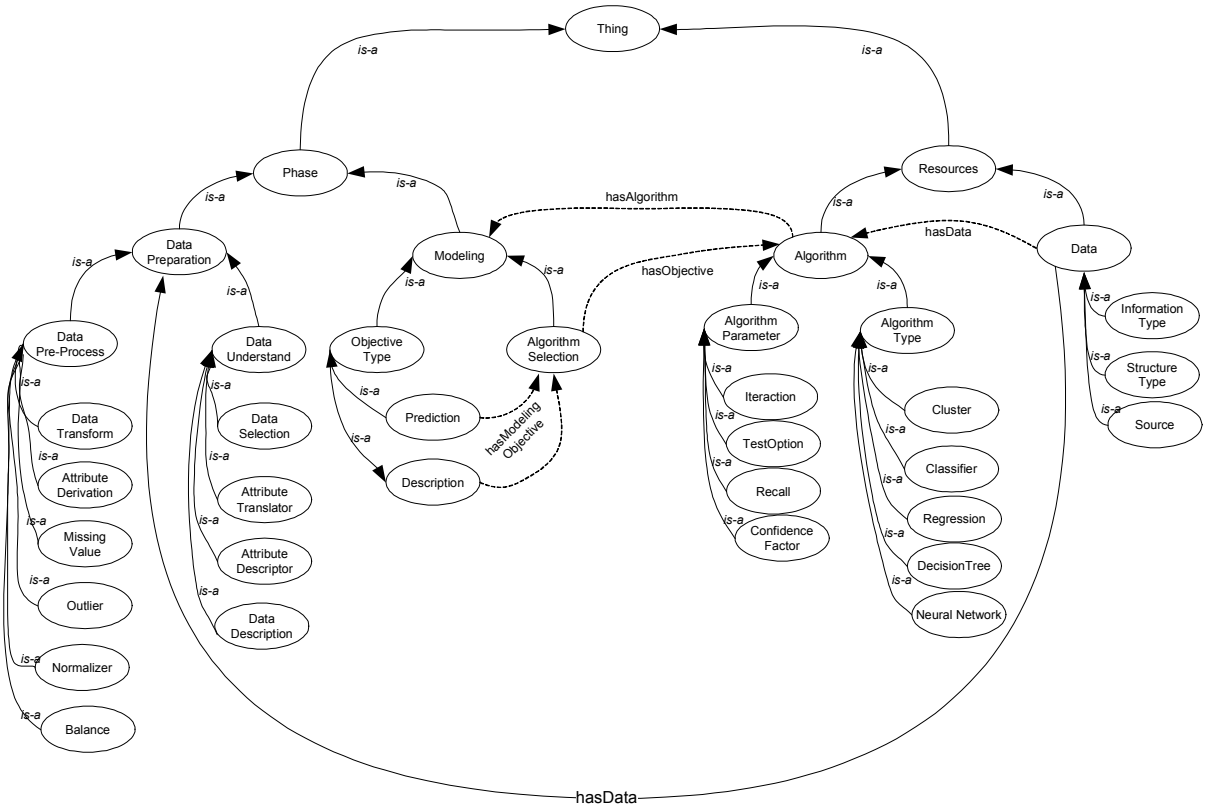


Figure 12: KDD ontology class-properties hierarchy general view

The KDD structure has two main distinct classes: resources and phase, as depicted in Figure 12 . The former, holds and refers to all assets used at KDD process, like data repositories or algorithms; the latter, refers to the practical development of KDD process phases, like data preparation or modeling. Each super class has its own subclass hierarchy. Moreover, there are relationships between each class (e.g., *hasData* or *hasAlgorithm*).

In the attachment section there is the KDD ontology OWL code (appendix 3); ontology class hierarchy (appendix 4), properties hierarchy (appendix 5) and protégé ontology print-screen (appendix 6).

4.1.4 An Ontology Proposal for Knowledge Discovery in Databases

Published in:

Journal of Sinoeuropean Engineering Research Forum

Volume 1 pages 34–39

ISSN 1757-4307

June 2008

Abstract:

To work efficiently data analysts need tools that enable accumulation, extraction and interaction of all the data and information about a particular problem. It is not clear how to integrate data that are poorly understood or lack unanimous support. Therefore, we need an approach that allows integration of the heterogeneous, diverse, distributed data and information with expert's knowledge. Currently, several advances in this field have been made, but most of them have certain shortcomings. In our opinion, ontologies are one of the most promising approaches to database marketing knowledge sharing, reuse and re-evaluation.

Through an exhaustive literature review we have achieve a set of domain concepts and relations between them to describe knowledge discovery in databases (KDD) process. Following methontology we had constructed our ontology in terms of process assistance role. Nevertheless, domain concepts and relations were introduced according some literature directives. Moreover, in order to formalize all related knowledge we have used some relevant scientific KDD and ontologies published works. However, whenever some vocabulary is missing it is possible to develop a research method in order to achieve such a domain knowledge thesaurus.

1. Introduction

Mining databases is challenging to analysts who has no domain expertise or vice-versa to domain professional who has no database exploration techniques expertise. Broadly, there are two kinds of knowledge that are involved in a knowledge discovery process: data mining based knowledge and domain knowledge. Data mining knowledge includes the knowledge about data mining algorithms, how they can be used, expected results

type, their requirements, parameters tuning, and formats of input data and so on. Domain knowledge includes the contextualization of the database with the business or objectives that lying the research project (e.g, marketing objectives or marketing activities), also includes the understanding of a dataset, relationships among variables, normal values to those variables, known casual relations or others relevant issues to the research.

Ontologies capture the domain concepts and their relations; therefore, it provides an alternative knowledge source than domain experts. There are many growing, large scale and shared ontologies which have been developed and used in various ways for helping the automation of knowledge discovery process (KDD). As example, there are ontologies to be used for feature generation in constructive induction - each generated feature corresponds to a concept created from the ontology; On other case, ontologies are used in order to reduce the amount of time required of a domain expert by staring with data in a database and inferring facts and relations about the variables - the systems scans new databases to obtain type and constraint information, the uses this information in the context of a shared ontology to intelligently guide the potentially combinatorial process of feature construction.

This work extents the KDD process to the ontology field in order to get some automation of the overall process. Despite the ontological knowledge approach to different research areas there's still a gap in its use in any domain application. Our goal is to exploit the KDD process itself in sharable and reusable knowledge base. Therefore a knowledge base with the results of each case will be populated. Specifically we hope to propose an ontology that, capture useful KDD process knowledge for reuse.

It is already well accepted that knowledge-base learning and discovery can be enhanced with automatic suggestion of some types of data or even some data mining models (Phillips and Buchanan, 2001). Mostly, however, the prior knowledge is specified separately for each new problem. Here we extend this line of research to design a general ontology based system that can capture prior knowledge found to be useful for one problem area and reuse it in another.

This paper is organized as follows. The next section mentions a brief ontological concepts background. The third section shows the research approach taken, following the proposed KDD ontology. At closing section discussion and conclusions are presented.

2. Ontologies

Ontology is a formal, explicit specification of a shared conceptualization (Jurisica *et al.*, 1999)(Gruber, 1995). Ontology might be a document or file that formally defines the relations among terms. The most typical kind of ontology has taxonomy and a set of inference rules (Dabholkar and Neeley, 1998). Any knowledge based system consists of at least two fundamental parts (using ontologies for scaffolding knowledge):*domain knowledge* and *problem-solving* knowledge. Ontology mainly plays a role in analyzing, modeling and implementing the domain knowledge (Staab and Studer, 2004).

Ontology is a key-enabling technology in order that it interweaves human understanding of symbols or terms with their machine process ability (using ontologies for scaffolding knowledge). Originally, ontology was developed in artificial intelligence to facilitate knowledge sharing and reuse. However, has become popular with different disciplines, such as knowledge management, natural language process and knowledge representation. The main reason to ontologies grown is "a shared and common understanding of a domain that can be communicated between people and application systems" (Gomez-Perez *et al.*, 2004).

Usually ontology is refined as "*specification of a shared conceptualization of a particular domain*". Ontology provides a shared and common understanding of a domain that can be communicated across people and application systems, and thus facilitate knowledge sharing and reuse. Also, aims at the machine-processing of information resources accessible to agents. Currently, the web is an incredibly large, mostly static information source.

Very shortly we describe here the main concepts found in an ontology languages:

- *classes* or concepts are the main entities of an ontology. They are interpreted as a set of individuals in the domain., e.g., Data or Algorithms. To each class it is possible to assign sub-classes, like DataType, or DataValueType for the class Data;
- *Instances or objects* are interpreted as particular individual of a domain, e.g, age it is an instance of the sub-class Demographics ;
- *relations* are the ideal notion of a relation independently to why it applies (e.g., the name relation in itself), they are interpreted as a subset of the products of the domain.
- *properties* are the relations precisely applied to a class (e.g., the gender of an individual);
- *property instances* are the relations applied to precise objects (the name of this individual)
- *datatypes* are a particular part of the domain which specifies values (as opposed to individuals),values do not have identities;

3. Research approach

Through an exhaustive literature review we have achieve a set of domain concepts and relations between them to describe KDD process.

Following METHONTOLOGY (Lopez *et al.*, 1999) we had constructed our ontology in terms of process assistance role. This methodology for ontology construction has five (Gomez-Perez *et al.*, 2004) main steps: specification, conceptualization, formalization, implementation and maintenance.

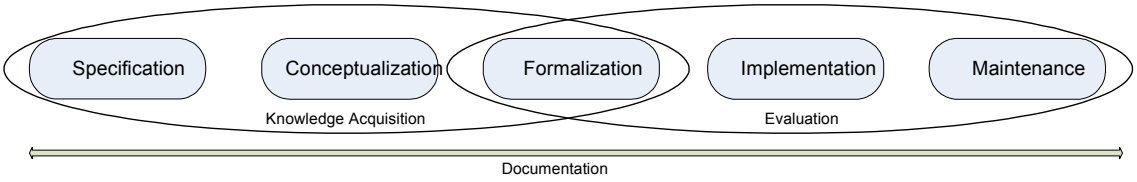


Figure 1: Methontology framework (adapted from [Lopez *et al.*1999])

Nevertheless, domain concepts and relations were introduced according some literature directives [Blazquez *et al.*1998][Smith and Farquhar2008]. Moreover, in order to formalize all related knowledge we have used some relevant scientific KDD [Quinlan1986] [Fayyad *et al.*1996, Fayyad and Uthurusamy1996] [Agrawal *et al.*1993] and ontologies [Phillips and Buchanan2001] [Nigro *et al.*2008] published works. However, whenever some vocabulary is missing it is possible to develop a research method in order to achieve such a domain knowledge thesaurus.

At the end of the first step of methontology methodology we have identified the following main classes (Figure 2):

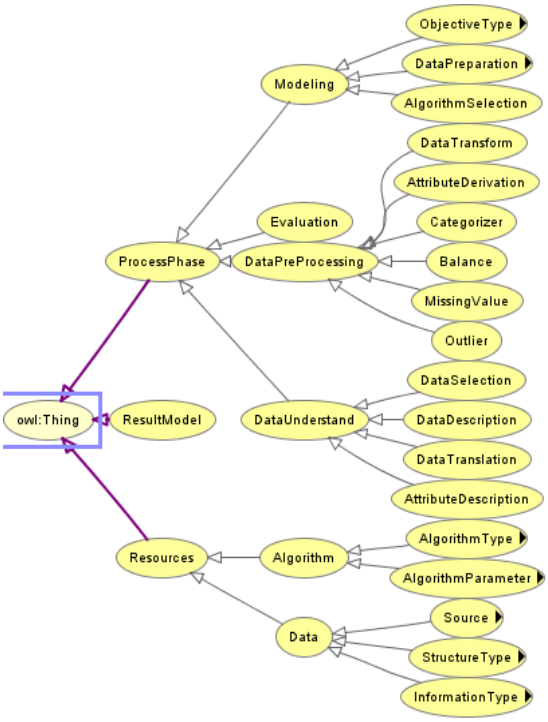


Figure 2: KDD ontology class taxonomy (partial view)

Our KDD ontology has three major classes: *Resource*, *ProcessPhase* and *ResultModel*. ProcessPhase is the central class which uses resources (Resource class) and has some

results (ResultModel class). The former Resource class relates all resources needed to carry the extraction process, namely algorithms and data.

The *ResultModel* has in charge to relate all KDD instance process describing all resources used, all tasks performed and results achieved in terms of model evaluation and domain evaluation. This class is use to ensure the KDD knowledge share and reuse.

Regarding KDD process we have considered four main concepts below the *ProcessPhase* concept (OWL class):

Data Understand focuses all data understanding work from simple acknowledge attribute mean to exhaustive attribute data description or even translation, to more natural language;

Data Preprocessing: concerns all data pre-processing tasks like data transformation, new attribute derivation or missing values processing;

Modeling: Modeling phase has in charge to produce models. It is frequent to appear as data mining phase (DM), since it is the most well known KDD phase. Discovery systems produce models that are valuable for prediction or description, but also they produce models that have been stated in some declarative format, that can be communicated clearly and precisely in order to become useful. Modeling holds all DM work from KDD process. Here we consider all subjects regarding the DM tasks, e.g., algorithm selection or concerns relations between algorithm and data used (data selection). In order to optimize efforts we have introduced some tested concepts from other data mining ontology (DMO) [Nigro *et al.*2008], which has similar knowledge base taxonomy. Here we take advantage of an explicit ontology of data mining and standards using the OWL concepts to describe an abstract semantic service for DM and its main operations. Settings are built through enumeration of algorithm properties and characterization of their input parameters. Based on the concrete Java interfaces, as presented in the Weka software API [Witten and Frank2000] and Protégé OWL, it was constructed a set of OWL classes and their instances that handle input parameters of the algorithms. All these concepts are not strictly separated but are rather used in conjunction forming a consistent ontology;

Evaluation and Deployment phase refers all concepts and operations (relations) performed to evaluate resulting DM model and KDD knowledge respectively.

Then, we have represented above concept hierarchy in OWL language, using protégé OWL software.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.semanticweb.org/ontologies/2009/5/DBMiPhDfpinto.owl">
 <owl:Class rdf:ID="InformationType">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Data"/>
  </rdfs:subClassOf>
 <owl:Class rdf:ID="Personal">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="lnformationType"/>
  </rdfs:subClassOf>
 </owl:Class>
 </owl:Class>
 <owl:Class rdf:ID="Demographics">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Personal"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
 <owl:Class rdf:about="#InformationType">
  <rdfs:subClassOf rdf:resource="#Data"/>
 </owl:Class>
```

Following Methontology, the next step is to create domain-specific core ontology, focusing knowledge acquisition. To this end we had performed some data processing tasks, data mining operations and also performed some models evaluations.

Each class belongs to a hierarchy (Figure 3). Moreover, each class may have relations between other classes (e.g., *PersonalType* is-a *InformationType* subclass). In order to formalize such schema we have defined OWL properties in regarding class' relationships, generally represented as:

Modeling^ has Algorithm(algorithm)

In OWL code:

```
 <owl:Class rdf:ID="AlgorithmSelection">
  <rdfs:subClassOf>
```

```
    <owl:Restriction>
     <owl:someValuesFrom rdf:resource="#Algorithms"/>
     <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasAlgorithm"/>
     </owl:onProperty>
    </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Modeling"/>
   </rdfs:subClassOf>
  </owl:Class>
```

The ontology knowledge acquisition, firstly, happens through direct classes, relationships and instances load. Then through the KDD instantiation, the ontology acts according to the semantic structure.

Each new attribute is presented to the ontology, it is evaluated in terms of attribute class hierarchy, and related properties that acts according it.

In our ontology Attribute is defined by a set of three descriptive items: *Information Type*, *Structure Type* and allocated *Source*. Therefore it is possible to infer that, Attribute is a subclass of *Thing* and is described as a union of *InformationType*, *StructureType* and *Source*.

At other level, considering that, data property links a class to another class (subclass) or links a class with an individual, we have in our ontology the example:

StructureType(Date)
→ *hasMissingValueTask*
→ *hasOutliersTask*
→ *hasAttributeDerive*

Attribute InformationType (Personal) & Attribute PersonalType(Demographics)
→ hasCheckConsistency
As example, considering the *birthDate* attribute, ontology will act as:
? Attribute hasDataSource

attribute hasDataSource (CustomerTable).
? Attribute hasInformationType:
Attribute hasInformationType (Personal) then:
attribute hasPersonalType(Demographics)
? Attribute hasStructureType
attribute hasStructureType (Date).
: attribute hasStructureType(Date) AND
PersonalType(Demographics) then:
: attribute (Demographics; Date) hasDataPreparation
: attribute (Demographics; Date) hasDataPreProcessing
AND Check missing values
AND Check outliers
AND Check consistency
AND deriveNewAttribute

In above example, the inference process is executed on reasoner for description logic (Pellet). It acts along both class hierarchy (e.g., *Personal or Demographics*) and defined data properties (e.g., *hasStructureType* or *hasDataPreparation*). In above example the attribute belongs at two classes: *Date* and *Demographics*. Through class membership, the *birthDate*, attribute inherits related data properties, such as *hasDataPreparation* or *hasDataPre-Processing*

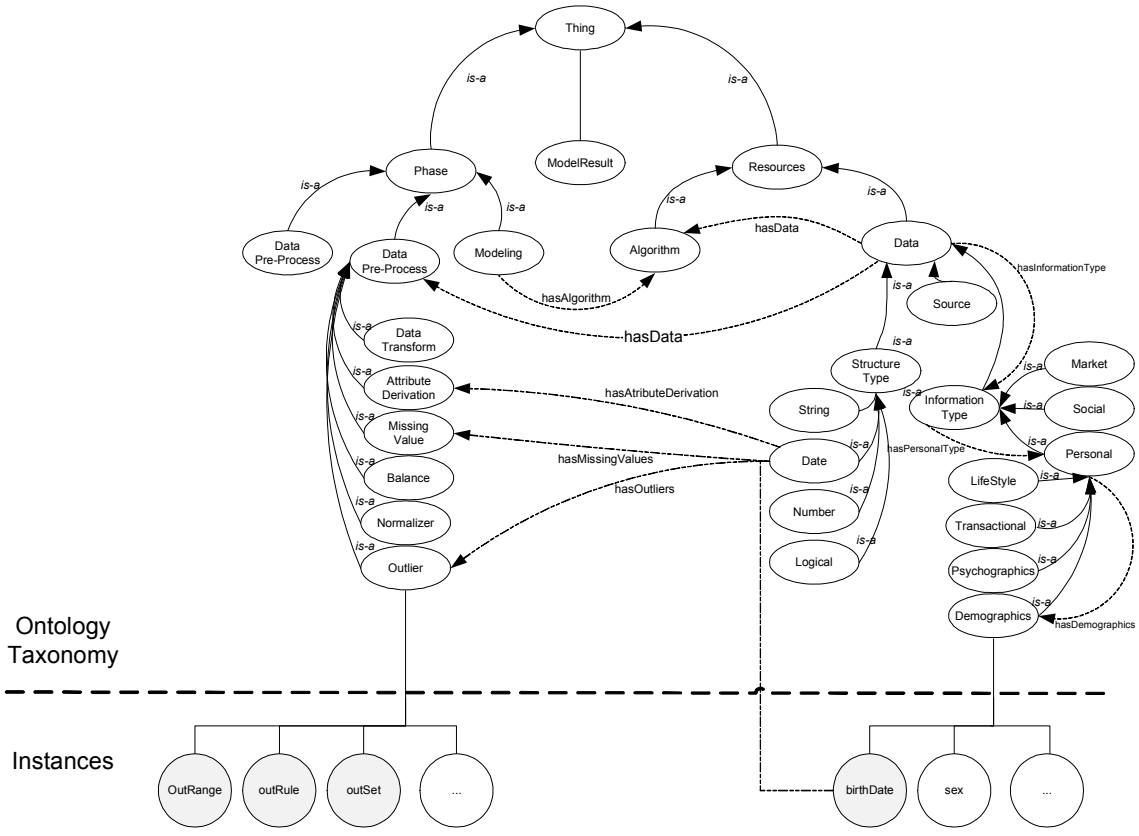


Figure 3: KDD class/property/instance relation example ilustration

4 Ontology Leaning cycle

Ontology assistance to KDD aims the improvement of the process allowing both better performance and extracted knowledge results.Since KDD process is the core competency of database use, it is the centre focus of our work.

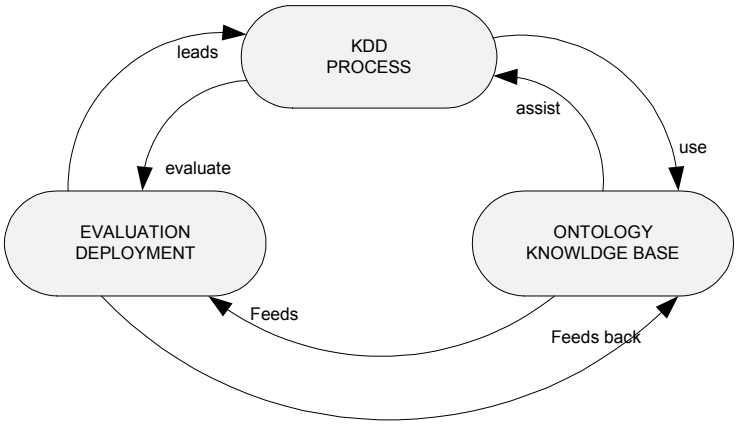


Figure 4: Ontology learning cycle

As depicted in Figure 4, KDD process is located at the centre of our system. Therefore, data analyst uses knowledge during the process execution; knowledge feeds performance for higher achievement, and performance leads measures performance through evaluation and deployment methods; performance feeds back knowledge (ontology update) for later use of that knowledge. Also knowledge drives the process to improve further operations.

Since the KDD process generates as output models, it was considered useful to represent them in a computable way. Such representation works as a general description of all options taken during the process. Based on PMML descriptive DM model we have introduced an OWL class in our ontology named *ResultModel* which holds instances with general form:

ResultModel {
domain Objective Type;
algorithm;
algorithmTasks;

algorithmParameters;

workingAlgorithmDataSet;

EvaluationValue;

DeploymentValue

}

Moreover, our ontology has the learning capability mutually assigned to aforementioned model the ontology structure. Then it is possible both: so suggest (e.g., algorithm) and rank each suggestion (e.g., accuracy). Such approach may lead in a future to the development of an automatic learning capability and is depicted in figure 4.

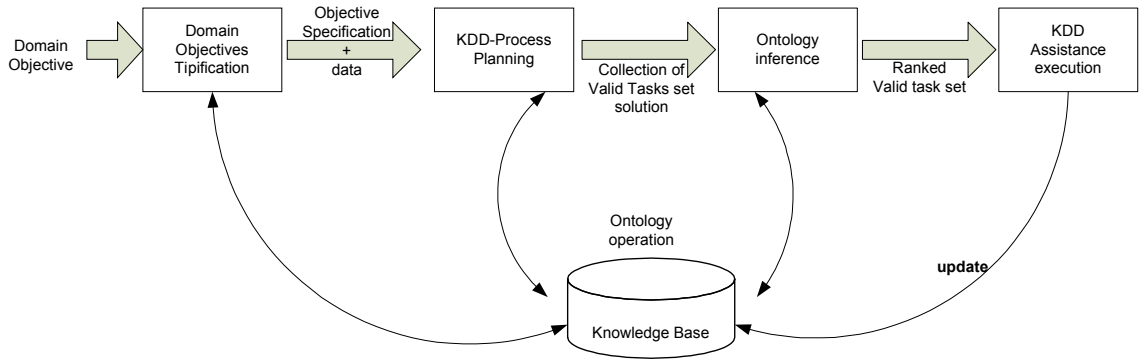


Figure 5: KDD ontology knowledge base operations

Data analyst is guided through the entire process supported by knowledge base. Such support is carried by domain objectives specification, KDD process planning, ontology inference or KDD assistant execution.

5. Conclusions

KDD is an inherently iterative process, and the proposed ontology may improve and accelerates the turn-around time between each phase and iterations. Such improvement derives from the ontology capacity to suggestion and recommendation.

This ontology is meant to be a subcomponent in the overall KDD process. Its usage of knowledge obtained from prior examples makes it applicable when several related databases are used.

Further work can be done in a variety of ways: this can be used for more specific knowledge extraction process or for more business oriented objectives.
We believe that this approach convincingly addresses a pressing KDD need.

References

[Dabholkar and Neeley1998] Dabholkar, P. A. and Neeley, S. M. (1998). Managing interdependency: a taxonomy for business-to-business relationships. Journal of Business and Industrial Marketing, 13:439–460.

[Gomez-Perez *et al.*2004] Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). Ontological engineering. Springer, 2nd edition.

[Jurisica *et al.*1999] Jurisica, I., Mylopoulos, J., and Yu, E. (1999). Ontologies for knowledge management: An information systems perspective. In for Information Sciences, A. S., editor, Proceedings of the Annual Conference of the American Society for Information Sciences (ASISâ€™99). American Society for Information Sciences.

[Lopez *et al.*1999] Lopez, M. F., Gomez-Perez, A., Sierra, J. P., and Sierra, A. P. (1999). Building a chemical ontology using methontology and the ontology design environment. IEEE Intelligent Systems Journal, 1:37–46.

[Phillips and Buchanan2001] Phillips, J. and Buchanan, B. G. (2001). Ontology-guided knowledge discovery in databases. In ACM, editor, International Conference On Knowledge Capture 1st international conference on Knowledge capture, pages 123–130. International Conference On Knowledge Capture.

[Staab and Studer2004] Staab, S. and Studer, R. (2004). Handbook on ontologies. International handbooks on information systems. Springer-Verlag.

4.2 Database Marketing Ontology

4.2.1 Introduction

In this section we present the developed work towards the database marketing ontology creation.

Marketing discipline is the application field for our research. In this work we intend to propose a DBM ontology. To the best of our knowledge there isn't any published scientific work focusing the DBM ontology creation. Therefore we needed to start from the scratch. Thus, we have based our work on a double methodological approach: Delphi method and ontological methodology 101. Firstly, we have collected all related DBM process knowledge and then we have pursued to the ontology construction.

4.2.2 Research approach

We have based this ontology construction on two different approaches: relevant related knowledge collection (Delphi method) and ontology construction (101 methodology).

Delphi method operation

Since we wanted to modulate DBM related knowledge counting with this expert panel contribution (Appendix 7 – Expert Panel to Delphi Method), we have started with an initial open question about DBM process and main objectives.

We have selected the three DBM common process structure among all answers. Also, we have selected the three common main objectives. Then we have created a form questionnaire where the selected DBM process and objectives were present and developed a new set of questions wondering to know how the panel realizes DBM in terms of marketing activities and does the panel expert group interpret the data, that is, how do they classify the data.

In this third interaction we have achieved the first evident division: from professional experts they interpret DBM as a tool and from academic experts understand DBM almost as a science. Here we have addressed a possible DBM framework. Also, we have introduced a possible marketing database knowledge structure regarding both professional and academic approaches. We used the expert panel's answers to comprehend how they perceive both DBM process and marketing databases structure.

Almost surprisingly, we have achieved a common acceptance from the expert panel. Indeed, they almost by unanimity had considered the proposed solution as definitive. Therefore, at the end of the Delphi process, in order to achieve our objective, we have introduced some smooth corrections, aiming to hear comments from the expert panel about marketing activities and data needed to reach them through DBM process, that is, we wanted to know how they perceive or make the connection between data and knowledge results from the DBM.

Throughout the expert panel feedback we have achieved as the following result:

- DBM process framework with six phases:
    - marketing objectives;
    - marketing activity selection;
    - data-based knowledge objectives;
    - data selection and preparation;
    - statistical or mathematics algorithms use (Data Mining);
    - data models evaluation and knowledge deployment.

- Main marketing data classification:
    - Information (different information type contents): personal (psychographics, demographics, lifestyle or transactional), market (economic, financial or social), and trigger (consumer, personal or society);
    - source: internal (from organization group, regarding active or legacy marketing systems) or external (rented databases or any other kind of information source)

- structure type (different types of data): date, number, character, set of values.

Methodology 101

We address this dilemma of ontology design and modeling. In particular, we consider conceptual modeling in the realm of process phases.

The fundamental objective behind conceptual modeling and that of ontology is the same – to conceptualize the domain of interest. Following our discussion in previous section we have considered that conceptual modeling holds the key to a comprehensive knowledge representation of a domain covering all aspects. Hence, we have selected the 101 methodology due its ontology lifecycle orientation and for the fact that we aimed to reuse the previously developed KDD ontology.

Ending this research we have formalized our DBM ontology, using the protégé OWL software (Appendix 6 – Protégé–Owl tool desktop).

4.2.3 Results

We have achieved a DBM domain ontology which focuses the entire process and related marketing knowledge.

4.2.4 Ontology Supported Database Marketing

Published in:

Journal of Database Marketing & Customer Strategy Management,
Palgrave&MacMillan,
volume 16, pages 76-91,
June 2009.

Abstract

Database marketing provides in depth analysis of marketing databases. Knowledge discovery in database techniques is one of the most prominent approaches to support some of the database marketing process phases. However, in many cases, the benefits of these tools are not fully exploited by marketers. Complexity and amount of data constitute two major factors limiting the application of knowledge discovery techniques in marketing activities. Here, ontologies may, nowadays, play an important role in the marketing discipline.

Motivated by its success in the area of artificial intelligence, we propose an ontology-supported database marketing approach. The approach aims to enhance database marketing with ontology by providing detailed step-phase specific information.

Our research work has its foundations in a double methodological approach using the Delphi and 101 ontology construction methodologies. Firstly, we use Delphi to structure related database marketing knowledge, then we align our work to the 101 methodology in order to systematize the knowledge extraction process and knowledge base creation.

The issues raised in this paper both respond and contribute to calls for a database marketing process improvement. Our work was evaluated in the relationship marketing domain focusing a relational marketing program database. The findings of this study not only advance the state of database marketing research but also shed light on future research directions.

1. Introduction

Database marketing (DBM) is a database oriented process that explores database information in order to support marketing activities and/or decisions.

The Knowledge Discovery from Databases (KDD) process is well established amongst the scientific community as a three phase process: data preparation, data mining and deployment/evaluation. This process is guided and controlled by both domain experts and database analysts. The KDD has been successfully applied in various domains, particularly in the marketing field.

Nevertheless, there seems to be a lack of knowledge concerning its application to different requirements and conditions, such as marketing objectives, available data, databases types or even missing domain expertise.

Our work focuses on the integration of knowledge extraction techniques within the database marketing discipline. Here, we introduce ontologies as a support to the knowledge structure and integration of both fields. In the context of knowledge sharing the term ontology means a specification of a conceptualization. This is, ontology is a description of the concepts and relationships that can exist for a single technological application or as a reference in a decision support system, and can be designed for the purpose of enabling knowledge sharing and reuse [Gruber1993, Jasper and Uschold1999, Zhou2007]. In this paper we provide an approach based on high-level abstraction using domain ontologies in order to construct a formal framework from data to marketing knowledge.

1.1 Current situation

Technology has provided marketers with huge amounts of data and artificial intelligence researchers with high level processing rate machines. Isolated practical DBM samples have been developed in different research fields [Payne and Frow2005, Ozimek2004, Kamakura *et al.*2003]. Also, there are some artificial intelligence projects that focus on marketing problems but their usage remains based on a single

methodology, e.g., algorithm performance analysis, data processing rates or data mining sample projects [Watada and Yamashiro2006] [Jans *et al.*2008]. This is, excluding proprietary business projects (marketing databases are normally used in a confidential environment), many of the research tasks (e.g., data preparation, data mining or evaluation phases) are focused to solve a specific problem without further inferences or information registration for other future cases or knowledge sharing.

1.2 Problem statement

Any time marketers need to develop DBM projects, they almost always start from scratch – much of the previous knowledge is unavailable or when available it is in an unhelpful format.

Much of the research developed in both fields (marketing and knowledge extraction techniques) focuses on DBM process and its results. Knowledge reuse in the marketing field is an innovation which could solve many of the practitioner's problems when developing its database based marketing activities.

1.3 Proposed solution

In computer science, ontologies provide a shared understanding of knowledge about a particular domain[Gruber1993]. Marketing ontologies although low in number are starting to come to the light through some marketing or computer research centers [Dabholkar and Neeley1998, Grassl1999, Bouquet *et al.*2002, Zairate *et al.*2006, Zhou *et al.*2006, Nogueira *et al.*2007]. Marketing ontologies are becoming more and more available and contribute to the understanding of the large amounts of data existent in the marketing field.

One of the promising possibilities for marketing ontologies is their use for guiding the process of knowledge extraction in DBM projects. A tool that gradually accumulates knowledge of the previous domain developed processes is appropriate due to its iterative nature. Researchers often rework their data in order to optimize further

interactions [Siqueira *et al.*2001]. Integrating this knowledge with ontology extends the usefulness of ontology.

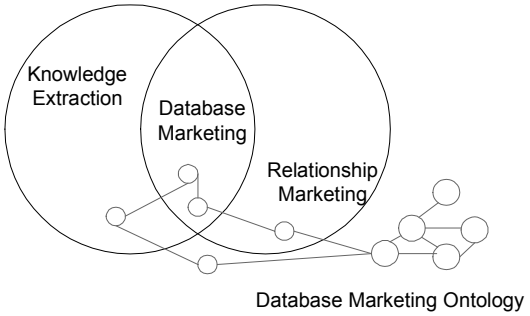


Figure 1: Database marketing ontology context

Therefore, the purpose of this work (Figure 1) is to focus on DBM as the intersection of two other disciplines (knowledge extraction techniques and marketing). It intends to capture main DBM concepts through knowledge discovery in databases and relationship marketing. The DBM Ontology (DBMO) should cover a semantic description of the supporting DBM process, comprising classified marketing objectives and activities, knowledge extraction methods, objectives and tasks.

The impact of this research is the future initiation of a shared DBM knowledge platform that will provide a trusted base between marketers, DBM practitioners and artificial intelligence researchers. Also, the ontology is intended to become the basis for future core ontology in the domain of DBM community.

This paper unfolds in the following manner: we start with the ontologies basis and knowledge issues in the marketing discipline then we outline the research approach. Research questions and research findings are presented in two subsequent sections. The results discussion is presented in section six, followed by conclusions and areas for potential further research.

2. Ontologies

Currently, ontologies are one of the most popular knowledge representation techniques. They have been proposed since the 18th century have been developed and deployed for sharable and reusable models. These ontologies aim to allow information modelling and knowledge management and reuse.

2.1. Ontology definition

Ontology is a description of conceptual knowledge organized in a computer-based representation [Nedellec and Nazarenko, 2005]. In artificial intelligence literature the most commonly quoted definition for ontology is "a formal, explicit specification of a shared conceptualization"[Gruber 1993]. A conceptualization, as it refers to an abstract model of one thing that describes the semantics of the data. An explicit specification means that the concepts and relationships in the abstract model are given explicit names (terms) and definitions (specification of the meaning of the concept or relation) that can be communicated amongst people and across application systems. Formal, due to how the meaning specification is encoded in a language which formal properties are well understood—in practice, this usually means logic-based languages that have emerged from the knowledge representation community within the field of artificial intelligence. Shared, means that the main purpose of ontology is generally to be used and reused across different applications and communities.

At a higher level ontology specifies the classes of concepts that are relevant to the application domain and the relations that exist between these classes. Ontology captures the intrinsic conceptual structure of a domain. For any given domain, its ontology forms the heart of the knowledge representation. Here, we very briefly describe what entities are found in an ontology language. These entities are mainly:

- Classes or concepts are the main entities of ontology. They are interpreted as a set of individuals in the domain, e.g., data or algorithms. It is possible to assign sub-classes to each class, like datasource, or datavaluetype for the class data;
- Instances or objects are interpreted as particular individual of a domain, e.g, age is an instance of the sub-class demographics;

- Relations are the ideal notion of a relation independently to why it applies (e.g., the name relation in itself), they are interpreted as a subset of the products of the domain.
- Properties are the relations precisely applied to a class (e.g., the gender of an individual); property instances are the relations applied to precise objects (the name of this individual)
- Datatypes are a particular part of the domain which specifies values (as opposed to individuals),values do not have identities;

Ontologies use a formal domain or knowledge representation agreed by consensus and shared by the entire community. There are several ways to represent such ontologies and many languages have been defined to represent them. There is a wide range of languages which goes from first-order logic (e.g., OWL or RDF) to frame-based languages implemented in ontology management systems (e.g., Protégé or Ontolingua).

3. Knowledge issues

Knowledge management is concerned with the representation, organization, acquisition, creation, use and evolution of knowledge in its many forms. In order to build effective technologies for knowledge management, we need to further our understanding of how individuals, groups and organizations use knowledge [Jurisica *et al.*1999, Mylopoulos *et al.*2004]. Currently more and more knowledge is represented in computer-readable forms, stressing the need to build tools that can effectively search databases, files, web sites to extract information, capture its meaning, organize and analyze it, and make it useful.

Ontologies are becoming more and more abundant in knowledge representation (KR) and management. Ontologies model the structure of data (classes and their properties or attributes), the semantics of data (in the form of axioms that express constraints such as inheritance relationships, or constraints on properties), and data instances (individuals). To integrate ontologies, we must understand the relationship between structures (classes and properties) and data (individuals) from different ontologies. Furthermore, we must

be able to use the semantics of ontology to model these relationships, and create a coherent and consistent integrated ontology [Udrea *et al.*2007].

3.1. Knowledge Representation and Ontologies

Knowledge Representation (KR) has long been considered one of the principal elements of artificial intelligence, and a critical part of all problem solving[Newell and level1982]. The subfields of KR range from the purely philosophical aspects of epistemology to the more practical problems of handling huge amounts of data[Guarino1995]. This diversity is unified by the central problem of encoding human knowledge - in all its various forms - in such a way that the knowledge can be used.

A KR must unambiguously represent any interpretation of a sentence (logical adequacy), have a method for translating from natural language to that representation, and must be reusable.

The central tenet of KR systems is a notation based on the specification of objects (concepts) and their relationships to each other. The main features of such a language are [Welty and Murdock2006]:

i. *Object-orientedness*. All the information about a specific concept is stored with that concept, as opposed, for example, to rule-based systems where information about one concept may be scattered throughout the rule base.

ii. *Generalization/Specialization*. Long recognized as a key aspect of human cognition, KR provides a natural way to group concepts in hierarchies in which higher level concepts represent more general, shared attributes of the concepts below.

iii. *Reasoning*. The ability to state in a formal way that the existence of piece of knowledge implies the existence of one other previously unknown piece of knowledge is important to KR.

iv. *Classification*. Given an abstract description of a concept, most KR languages provide the ability to determine if a concept fits that description or not. This is actually a common special form of reasoning.

KR systems have some limitations when dealing with procedural knowledge. As example of procedural knowledge [Welty1996] would be Newton's Law of Gravity - the attraction between two masses is inversely proportional to the square of their distances from each other. Given two bodies, with slots holding their positions and mass, the value of the gravitational attraction between them cannot be inferred declaratively using the standard reasoning mechanisms available in KR languages. Still, a function or procedure in a programming language could represent the mechanism for performing this "inference" quite well. Ontologies can deal with this kind of knowledge by adding a procedural language to its representation. Therefore, the knowledge is not being represented in a declarative way; it is being represented as C or LISP (computer programming languages) code which is accessed through a slot. This is an important distinction - there is knowledge being encoded in those computer programming functions that is not fully accessible. The system can reason with that knowledge, but not about it – here the ontological role.

Ontologies are a key part of a broader range of semantics based technologies which include the areas of KR and automated inference that arose within the artificial intelligence community [Jasper and Uschold1999, Uschold and King1995]. Many different representation formalisms have been explored, and reasoning engines developed. In strict sense, ontologies may be considered as a sub-area within KR [Uschold and Gruninger2004], since almost every knowledge base frequently has ontology as its main backbone. Ontologies capture the intrinsic conceptual structure of a domain.

The focus on knowledge sharing and reuse constitute the major difference between ontologies and KR in general. Moreover, ontologies go beyond KR limits, since they are designed to allow reasoning activities.

4 Ontologies in the context of database marketing

This research on marketing ontologies is part of a larger project that deals with the extraction of marketing knowledge from large and heterogeneous marketing databases. Thus we need a tool for knowledge representation, reasoning and decision support.

Here, ontologies role in DBM has particular significance as they focus on a crossover of areas. That is, to develop DBM, both marketing and extraction techniques knowledge is needed. Thus, ontologies can play an important role describing in a semantic form all concepts and techniques around the process. Moreover, with such a description it will also be possible, in a second phase, to introduce metrics in order to compare and therefore select and suggest the best approaches and methods in the context of a new project.

Ontologies should provide consensual knowledge about a certain domain or area interchangeable by the community. Such ontologies would allow common applications to be developed due to their compatible formats. In this work we are not designing a global marketing ontology representing all varied aspects of the marketing domain. We are proposing domain ontology as an integral part of a global marketing system. Our ontological proposal deals with some marketing knowledge and extraction process methods and tasks necessary to the DBM process and thereafter for marketing ontology. According to some researchers this ontology is classified as application ontology [Sowa2000] serving our main global project. As such, we focus only the study of DBM related concepts.

Ontologies are also like conceptual schemata in database systems. A conceptual schema provides a logical description of shared data, allowing application programs and databases to interoperate without having to share data structures. While a conceptual schema defines relations on data, ontology defines terms with which to represent knowledge [Zhou *et al.*2006]. For present purposes, one can think of data as that expressible in ground atomic facts and knowledge as that expressible in logical sentences with existentially and universally quantified variables. Ontology defines the vocabulary used to compose complex expressions, such as, those used to describe

resource constraints in planning problems. From a finite, well-defined vocabulary one can compose a large number of coherent sentences. That is one reason why vocabulary, rather than form, is the focus of specifications of ontological commitments

In computer science, ontologies have appeared in a variety of forms, ranging from lexicons, to dictionaries and thesauri or even first order logical theories. Lexicons provide a standardized dictionary of terms for use during, e.g., indexing or retrieval. Dictionaries can be organized according to specific relations to form hierarchies (taxonomies, meronomies, etc.). Thesauri add related terms to any given term. In DBM as in any of these forms, ontologies are useful because they encourage standardization of the terms used to represent knowledge about a domain. When ontologies are formalized in first-order logic, they can also support inference mechanisms [Mylopoulos *et al.*2004]. For a given collection of facts, these mechanisms can be used to derive new facts or check for consistency. Such computational aids are clearly useful for knowledge management, especially when one is dealing with complex problems or handling large amounts of knowledge.

The essence of our marketing ontology is a collection of DBM process-relationships from marketing row data to marketing knowledge. The basic facts we need to represent are of the form that a specific classification of marketing activities, data used and knowledge extraction process techniques adopted. As example, "a married man with children buys beer and diapers during world football cup." The challenge is to find a representation of this kind of knowledge in a convenient and economical way that fits into our DBM ontology framework.

5. Research approach

Due to the nature of the research we split the project into two sections adopting different research approaches for each one: Delphi and 101 ontology construction approaches.
Firstly, we focus our work on marketing knowledge, in order to structure its main concepts and systematize the overall organization, namely, marketing objectives and activities and main marketing database data types. Secondly, in order to design and

improve overall DBM perspective we focus the process on the semantic description of used procedures and methods in order to systematize the knowledge extraction process and knowledge base creation.

Our work ends by pointing to a possible framework that will lead future DBM projects supported by ontologies.

5.1. Delphi methodology

The Delphi method is normally used to structure a group communication process to deal with and to build consensus about a particular and complex topic. The method works based on an expert panel group (anonymous experts - no expert knows who is else is on the panel) who answer to proposed questions and formulate a set of hypotheses about it [Linstone and Turoff2002] [Chu and Hwang2008]. Then, the method is developed on the dialectical inquiry approach: the researcher introduces a set of questions in order to establish an opinion or view from the expert panel. Then the expert panel (individually) answers reporting a formulation (conflicting opinion or view). The researcher in charge has to generate a synthesis (a new agreement or consensus) and submit it again to the expert panel. This loop only ends when the researcher achieves a consensus with all expert panel members.

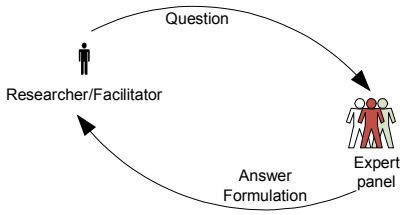


Figure 2 Delphi methodology process

Nowadays, Delphi is considered a useful method for eliciting and aggregating expert opinion whenever there is a lack of viable or practical statistical techniques. It can be defined as a medium-term qualitative forecasting method that is based on building a consensus amongst a group of experts [Armstrong2006]. A Delphi type study enables an exchange of information amongst experts over a number of rounds (iterations) and allows experts to react to the information gathered during each round and to fine-tune their forecast by means of a feedback mechanism (controlled retroaction). Beyond these

three main principles (anonymity – iteration – retroaction), the method's validity is firstly based on a rigorous selection of experts whose combined knowledge and expertise must reflect the full scope of the problem area.

Some authors have suggested asking the persons involved to estimate their own degree of expertise; with others considering that the level of expertise does not necessarily need to be high[Rowe and Wright2001]. Delphi's validity is also dependent on the size of the group of experts [Vernette1997] (research suggests, that the minimum threshold is 5–7 experts, and that a range of 8–10 offers the best precision/cost ratio. Beyond 12 experts, information contributions are marginal). The method's validity relies on a strict implementation of the process: three iterations are usually needed to obtain a satisfactory consensus [Armstrong2006].

Methodology 101

101 Methodology is based on the principle that, there are several viable alternatives for ontology development and ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.

Ontologies have become core components of many large applications. This methodology presents a set of tasks for creating ontologies based on declarative knowledge representation [Noy and McGuiness, 2003]. It leverages the author's experiences developing and maintaining ontologies in a number of ontology environments including Protégé, Ontolingua, or Chimaera. The Ontology 101 methodology is relatively simple, since it defines simple and, some of them, generic steps. Indeed this methodological approach does not make assumptions about knowledge representation/ontology language.

Figure 3: 101 methodology framework (adapted from [Noy, 2003])

This methodology uses the ontology domain and scope (based on related knowledge) to pragmatically determine which the best approach for ontology development is. Also, this methodology assumes the ontology development is a process of iterative design that will likely continue through the entire lifecycle of the ontology.

Ontology Development Process

The ontology development process refers to which activities are carried out when building ontologies. 101 methodology uses a generic approach from the domain application to the effective instance creation, through the following steps [Noy and McGuiness, 2003]:

- *Determine the domain and scope of the ontology*, throughout the answer to some basic but relevant questions, like what is the domain that the ontology will cover;
- *Consider reusing existing ontologies*: since it is almost always worth considering what someone else has done and checking if we can refine and extend existing sources for some particular domain and task;
- *Enumerate important terms in the ontology* throughout an extensive list of all domain related knowledge terms, with which it would be useful either to make statements about or to explain to a user;
- *Define the classes and the class hierarchy*: There are several possible approaches in developing a class hierarchy [122] [63]: top-down; middle-out and bottom-up;
- *Properties definition*: Once some classes have been defined, it must be identified the internal concept structure – properties. All subclasses of a class inherit the slot of that class, e.g., all slots of the class *Data* will be inherited to all subclasses of *Information Data Type* , including *Source* and *Structure Type*;
- *Define the facets of the slots (restrictions)*: slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take;

- *Create instances:* this last step consists in creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires (i) choosing a class, (ii) creating an individual instance of that class, and (iii) filling in the slot values.

6. Research questions

The framework for this project was conceived from different research area literature review: relationship marketing [Coviello and Brodie1998, Coviello *et al.*2006], database marketing [Ozimek2004, Coviello *et al.*2001, Wehmeyer2005], ontologies [Dabholkar and Neeley1998, Grassl1999, Diamantini *et al.*2006, Zhou *et al.*2006] and knowledge discovery in databases [Fayyad *et al.*1996, Phillips and Buchanan2001, Buckinx and den Poel2005, Buckinx *et al.*2007].

To define the expert panel we focused on the individual's reputation and recognition in academic and business circles. To avoid any type of collusion or friendship side effects, we did not ask for experts' names but devised a questionnaire on practitioners and researchers. Then we sent the questionnaire to each one of them. We expected to know their opinion from the answers to the questions.
Through Delphi methodology we have started from relationship marketing field attaining to construct a knowledge tree where main objectives, action programs and related activities are identified.

According to these first stage objectives we proposed the following questions for discussion by our expert panel:

i) Regarding relationship marketing context which are main marketing activities that use DBM approach?
ii) Regarding relationship marketing context which are main DBM objectives?
iii) Which is the main type of data used in DBM projects? After constructing the marketing knowledge structure tree, we proceeded with Action Research methodology that led us to the answers to the following main questions:
    a) Principal marketing database data type information;
    b) Main DBM steps from data to customer knowledge;

c) Operational DBM matrix aligning knowledge extractions methods and marketing activities and objectives.

Research cycles from both methodologies, in combination with the reconnaissance of the expert panel (first phase) and professional marketers (second phase), led to the development of the final framework of DBM process supported by ontologies and knowledge discovery in databases. The proposed framework has the capacity to suggest solutions from previous knowledge registered in the knowledge base.

7. Research Findings

As referred previously, this first phase work was developed according to Delphi methodology. We sent the questionnaire to each one of them. From each one of them we expected to know his/her opinion from their answer to the question. This interaction took place during five cycles. That is, there were four iterations before we considered (common agreement about the subject) all the answers to the proposed questions to be stable.

Our findings at this stage are summarized in the following table (Table 1).

Table 1 Delphi method findings

Research issue	**Findings about the research issues**
Regarding relationship marketing context which are the main marketing activities that use DBM approach?	Same marketing activities may be developed under different marketing disciplines, e.g., customer identification, can be developed both in relationship marketing program as well as in direct marketing. That is, there's a non-exclusive set of possible marketing activities available where DBM projects took place. Aligning with relationship marketing objectives we have organized as follows [Peppers and Rogers1999]: *To identify* - Customer knowledge or identification - Customer needs - Customer wants *To differentiate* - Customer segmentation - Customer categorization

	- Customer profiling *To interact* - Cross and Up-selling - Cross marketing - One-to-one marketing - Customer reactivation *To customize* - Customer loyalty acquisition - Customer fidelization - Customer affiliation
Regarding relationship marketing context which are the main DBM objectives?	DBM process is aligned with the marketing activity which holds its context. Therefore amongst the proposed DBM objectives we have organized the following as main objectives : - Segmentation - Classification or clustering - Market basket analysis - Prediction future behaviour - Description - Churn - Reactivation
Which is the main kind of data used in DBM projects?	Both literature and expert panel suggest that the information gathered in marketing databases is mainly organized or well defined as the following data types (some examples of each one are presented): Psychographics: personal data that can easily be changed. - monthly income - professional occupation - scholarship Demographics: physical and personal data that is almost definitive and almost never changes. - gender - marital status - birth date - children - race Transactional: consumer based information regarding its commercial activity - monthly consumption - number transactions/month - number items/month - shops visited - promotional acceptance

Lifestyle or behaviour: consumer or social related information.

- hobbies
- car type
- holidays
- club membership

In addition to the above customer oriented data types there are two other groups of data:

Market data: environmental market data

- Financial (e.g., inflation tax rate)
- Market (e.g., market or product share)
- Social (e.g., national birth, death or other sensus)

Trigger events data:

- Consumer (e.g., married status change or children number)
- life related (e.g., new car or new house)
- others (e.g., accident, prison, tax penalties)

Following the previous Delphi methodology research which has given us a marketing knowledge concepts structure tree, we proceed with 101 methodology. At this point we aim to answer the following main questions:

i) Principal marketing database data type information;

ii) Main DBM steps from data to customer knowledge;

iii) DBM matrix: marketing activities objectives, knowledge discovery type models and marketing data type connection.

We have developed 101 ontology construction method research at two simultaneous theoretical and practical levels and therefore two working focus groups:

i) practice over a real relationship marketing program database;

ii) literature oriented field research (an expert panel had explored scientific literature and achieved a set of possible tracks to each one of the research focus).

Data description was collected through both focus groups. Because the research phenomenon is contemporary and no prior research has been conducted or was known at the time this paper was written, both of them had collected DBM process descriptions as well as knowledge discovery approaches. These focus groups were interesting

because they generated insights [Carson *et al.*2001] from both DBM practice and process knowledge (namely at data preparation and pre-processing levels).

Table 2: 101 intermediate findings towards ontology construction

Research issue	**Findings about the research issues**
Principal marketing database data type information	From some literature review and supported by previous Delphi methodology - Psychographics - Demographics - Lifestyle - Transactions
Main DBM steps:	Based on both practice and literature review we have considered the following steps as a stable DBM process framework: - Marketing objectives definition - Data selection - Data preparation - Data pre-processing - Modelling - Deployment - Evaluation
DBM matrix	<table><tr><td></td><td>Marketing Objectives| Activities</td></tr><tr><td>Knowledge Methods</td><td>Description set: { Data set Data selection Data pre-processing Data preparation Algorithm used Technical evaluation Business evaluation }</td></tr></table>

In combination with the focus groups, convergent work was used to further enumerate terms, define classes, properties and restrictions and refine the aimed theoretical framework. Convergent work involves, for example, transposing from reviewed literature approaches or panel suggestions for the practical domain. Each interaction was then registered in terms of type of data, data analysis algorithm used and results achieved with them. Convergent work involves also conducting a series of in-depth

working groups in order to explore other insights that were not previously registered. That is, the process is very structured and only ends when no new information remains uncovered or unregistered.

Ending the 101 ontology construction a practical and functional analysis was made towards a possible conceptual semantic map. Turning to analytic generalization, we can then build a theoretical framework [Yin2003] linked to extant literature that shows how the DBM process is developed, how associated marketing knowledge can be structured and which knowledge discovery approaches may be used. Our research allows the identification of three main components of the DBM process (Figure 4): inputs (marketing objectives, marketing activities and marketing data), tasks (data handling and data modeling), and outputs (evaluation, deployment and business value).

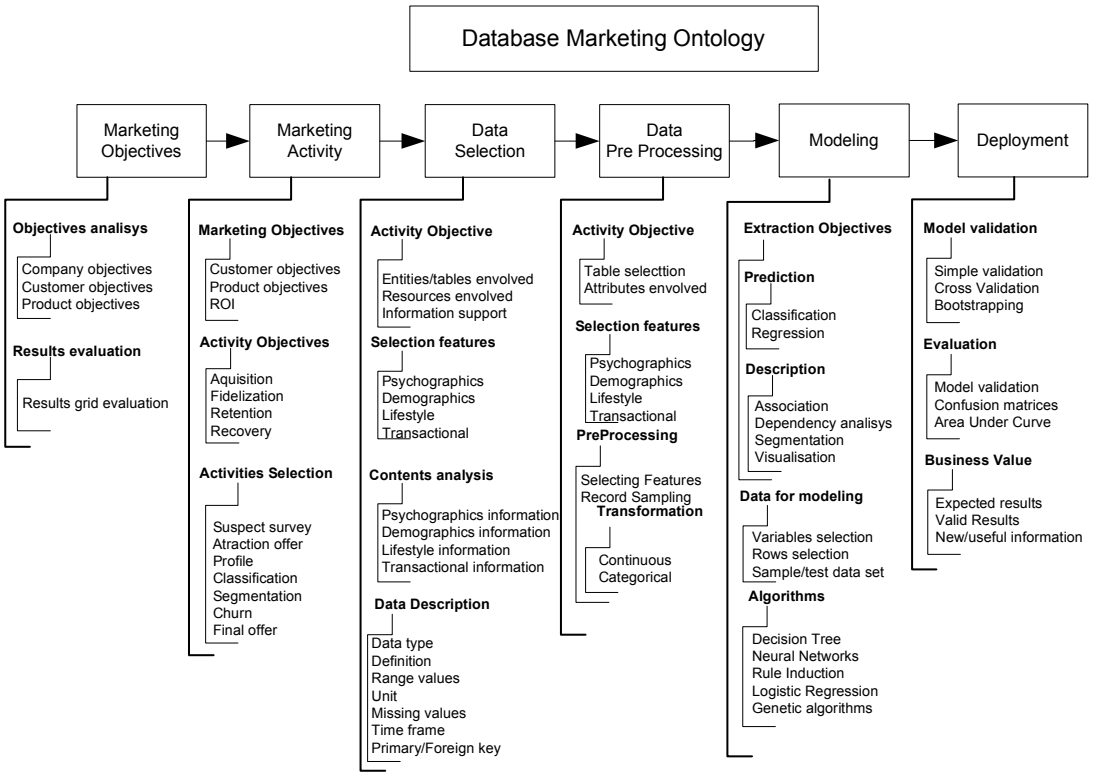


Figure 4 DBMO architecture

The ontological commitment as form of a matrix evaluation whereas data loaded, tasks and methods taken and results obtained are evaluated and registered in a knowledge base.

Knowledge base: { Results={DBMimodels[{input}{tasks}]} }

In order to test and verify the knowledge consistency and therefore the knowledge structure we have collected a large amount of relationship marketing data from a multinational distribution company. Our database contains at an individual level different kinds of marketing information, such as demographics, psychographics, lifestyle and transactional information. Also, some external data is presented as example market or as financial information.

We have processed the data using WEKA [Witten and Frank2000], free data mining software and we have found different results according to different data and algorithms used (Table 3). Therefore, we extracted information and organized it on an individual perspective.

Table 3 DBM process example

Case 1		
Marketing objective: customer profile		
Data:	Personal-psychographics	
	birthDate	
	gender	
	children	
	incomePerCapita	
Personal-Demographics		
	maritalStatus	
	houseHoldDimension	
Personal-Transaction		
	customer id;	
	productConsumption_1	
	productConsumption_2	
…		
	productConsumption_128	
	supermarketMonthlyConsumption	
Individuals:	613 000	
cleaned records:	64 000	
Data preparation tasks used:		missing values;
		duplicationSelector;
		unitDeviations;
outliers.		
Data transform tasks:		matrizTranspose;
		discretization.
Data Mining Method:		Classification
Algorithms:		SOM
		C 5.0
Evaluation		pccConfusionMatrix

All information regarding each developed DBM project has been registered in a knowledge base table which has information as follows (Table 4).

Table 4 Knowledge base table record example

{
marketing objectives;
marketing activity;
data used [{demographics}, {psychographics}, {life style},{ transactional}];
data quality[{outliers},{missing values},…]
data procedures [{selection},{preparation},{pre-processing}]
algorithms used [{clusterers}, classifiers}, neuralNetworks}, geneticAlgorithms}, statistical}…]
evaluation method [{auc}, {pcc}…]
}

To classify the degree of success of a DBM project is very subjective. Nevertheless, according to our approach we can perform, register and implement some analytical procedures that will lead to some DBM evaluation. Within this research we assume data mining evaluation models like AUC (area under curve), confusion matrix or principal components analysis are used. For each model we also evaluate which kind of data was used and related quality in terms of completeness, outliers and missing values. Regarding each data set used, we have registered all data tasks performed, like data cleaning, data transformation or data reduction. Related to the modelling phase a table was created in order to not only register which algorithms were performed but also which data from loaded data sets was used.

The model deployment is performed on two counts: analytical deployment and business perspective: i) analytical deployment: focusing the algorithm performance; ii) business perspective: regarding its practical application, that is, there are models with high accuracy but with low interest (e.g., a rule like all women buy female products) and others with low rating but with high impact regarding business value (e.g., customers aged under 50 years, two children, married, high level occupation have a 50% probability of buying your product).

8. Discussion

One of the promising interests of DBM ontology is its use for guiding the knowledge extraction process from marketing databases. This idea seems to be much more realistic now that semantic web advances have given rise to common standards and technologies for expressing and sharing ontologies [Coulet *et al.*2008, Smith *et al.*2008].

In this way DBM can take advantage of domain knowledge embedded in DBMO:

i) at the marketing activity definition, ontology can indicate a global perspective which is possible to do or not to do with the available resources, e.g., based on data completeness or heterogeneity;

ii) from a DBM objectives point of view, ontology may suggest or select the most appropriate approaches to treat the available data;

iii) during the data preparation step, DBMO can facilitate the integration of heterogeneous data and guide the selection of relevant data to be used;

iv) at the modelling phase (e.g. data mining), domain knowledge allows specification of constraints to guide data mining algorithms by, e.g., narrowing search space;

v) during the interpretation step, domain knowledge helps experts to visualize and validate extracted units.

Therefore, using a general framework it is possible to illustrate a general perspective of how the system works (Figure 5). We have considered a three layer architectural approach:

- Physical layer, which holds the process development tasks, namely data handling (selection, preparation, pre-processing and transformation) and modeling;
- Ontological layer acts like a guide to the data analyst and as a reference to the marketer expert;
- Presentation or user layer, has the interaction role amongst above layers and users.

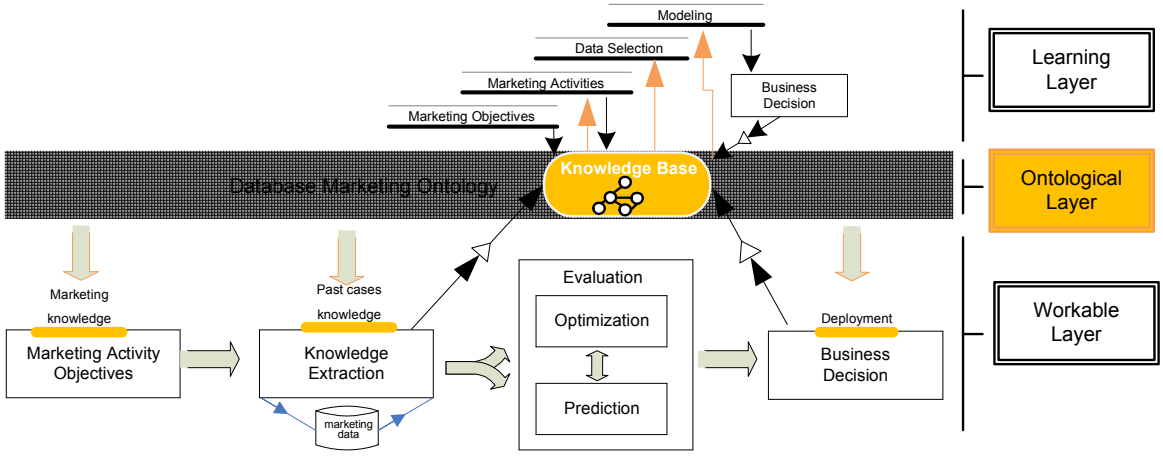


Figure 5 DBMO general framework

DBMO divides the DBM process into four main phases: Marketing activity objectives, knowledge extraction, evaluation and business decision.

With this research we can suggest some general roles for the ontology in each DBM phase:

- *Marketing activity definition*: The role of ontologies in business understanding is not peculiar to the marketing discipline. Domain ontologies are an important vehicle to inspect a domain prior to committing to a particular task. Semi-formal ontologies can help a newcomer to get familiar with most important concepts and relationships, while formal ontologies allow the identification of conflicting assumptions that might not be obvious at first sight;
- *Knowledge extraction*: For improved data exploration, elements of ontology have to be (presumably manually) mapped onto elements of the data scheme and vice versa. This will typically lead to selecting a relevant part of ontology (or multiple ontologies) only. Another relevant issue is the connection between the *Data Preparation* phase and the subsequent *Modeling phase*. Concrete use of domain ontology depends partially on the chosen mining tool/s. Ontology may characteristically help by identifying multiple groups of attributes and/or values according to semantic criteria. In the *Modeling phase*, ontologies might help design the individual mining sessions. In particular for large datasets, it might be worthwhile to introduce some ontological bias, e.g., to skip the quantitative

examination of hypotheses that would not make sense from the ontological point of view, or, on the other hand, of two obvious ones;

- *Evaluation phase:* the discovered model/s have the character of structured knowledge built around the concepts (previously mapped on data attributes) and can be interpreted in terms of ontology and associated background knowledge;
- In the *Business Decision* phase: extracted knowledge is fed back to the business environment. Provided we previously modeled the business using ontological means, the integration of new knowledge can again be mediated by the business ontology. Furthermore, if the mining results are to be distributed across multiple organizations (say, using the semantic web infrastructure), mapping to a shared ontology is inevitable.

9. Conclusions

The extent, degree and speed of communication enabled by the ontology makes it a synergistic component of DBM strategy. Our proposed DBMO, an ontological DBM approach solution appears promising for both marketers and computer scientists.

The results of this research have implications for both theory and practice. Related to practice, the very first one relates to the possible feedback among different DBM projects depicted in a table with all used resources registered. This enables the construction of a knowledge base containing suggestions or work profile capability. According to the previous registered experiments, the knowledge base will be capable of indicating for each marketing objective which marketing activities, data and also the tasks that should be carried out.

Another implication relates to the benefits of a global view of marketing databases' role in marketing objectives. There is only one way to have a successful DBM project: it must have appropriate data type and quality.

The research findings and contributions have several implications for the theory about ontologies and DBM, as well as for the integration of research methodologies such as Delphi and 101 ontology construction. This research provides new insights into DBM theory in two ways:

i. It appears to provide the first global investigation about the intersection of ontologies and DBM in organizations and how it may be achieved. This research contributes to the theory-deficient area of the integration of ontologies and DBM, providing the first approach to a theoretical framework for such a phenomenon;
ii. There is little literature dedicated to marketing ontologies and thus this research appears to be the first academic investigation of this phenomenon.

The DBMO model further emphasizes the importance of the marketing knowledge being structured in order to allow resource reuse or even to achieve synergies in marketing activities development. Thus managers and marketers should be aware of this issue, because there is a loop through which performance of DBM process can effectively be improved.

This research showed that the most important output of the ontological approach is an enabling of effective DBM assistance without in-depth expertise in e.g., data mining tools. Supported by the knowledge base ontology is capable of suggesting the pathway from data to desired knowledge.

References

[Armstrong2006] Armstrong, J. S. (2006). Findings from evidence-based forecasting: Methods for reducing forecast error. International Journal of Forecasting, 22(3):583–598.

[Baskerville1999] Baskerville, R. L. (1999). Investigating information systems with action research. Communications of AIS Volume 2, Article 19, 2:19–51.

[Bouquet *et al.*2002] Bouquet, P., Dona, A., Serafini, L., and Zanobini, S. (2002). Contextualized local ontology specification via ctxml. In for Artificial Intelligence, A. A., editor, MeaN-02 AAAI Workshop on Meaning Negotiation, Edmonton, Alberta, Canada. AAAI.

[Buckinx and den Poel2005] Buckinx, W. and den Poel, D. V. (2005). Customer base analysis: Partial defection of behaviorally-loyal clients in a non-contractual fmcg retail setting. European Journal of Operational Research, 164 (1):252–268.

[Buckinx *et al.*2007] Buckinx, W., Verstraeten, G., and den Poel, D. V. (2007). Predicting customer loyalty using the internal transactional database. Expert Systems with Applications, 32:125–134.

[Carson *et al.*2001] Carson, D., Guilmor, A., Perry, C., and Gronhaug, K. (2001). Qualitative Marketing Research. Sage.

[Chu and Hwang2008] Chu, H.-C. and Hwang, G.-J. (2008). A delphi-based approach to developing expert systems with the cooperation of multiple experts. Expert Systems with Applications, 34:2826–2840.

[Coulet *et al.*2008] Coulet, A., Smail-Tabbone, M., Benlian, P., Napoli, A., and Devignes, M.-D. (2008). Ontology-guided data preparation for discovering genotype-phenotype relationships. Journal of BMC Bioinformatics, 9:1–9.

[Coviello and Brodie1998] Coviello, N. and Brodie, J. (1998). From transaction to relationship marketing: an investigation of managerial perceptions and practices. Journal of Strategic Marketing, 6(3):171–186.

[Coviello *et al.*2001] Coviello, N., Milley, R., and Marcolin, B. (2001). Understanding it-enabled interactivity in contemporary marketing. Journal of Interactive Marketing, 15(4):18–33.

[Coviello *et al.*2006] Coviello, N., Winklhofer, H., and Hamilton, K. (2006). Marketing practices and performance of small service firms - an examination in the tourism accommodation sector. Journal of Service Research, 9(1):38–58.

[Dabholkar and Neeley1998] Dabholkar, P. A. and Neeley, S. M. (1998). Managing interdependency: a taxonomy for business-to-business relationships. Journal of Business and Industrial Marketing, 13:439–460.

[Diamantini *et al.*2006] Diamantini, C., Potena, D., and Smari, W. (2006). Collaborative knowledge discovery in databases: A knowledge exchange perspective. In Fall Symposium on Semantic Web for Collaborative Knowledge Acquisition, pages 24–31, Arlington, VA, USA. AAAI, AAAI.

[Dick2008] Dick, B. (2008). Postgraduate programs using action research in action learning, action research and process management: Theory, practice, praxis. Technical report, Faculty of Education, Griffith University.

[Fayyad *et al.*1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. In Magazine, A., editor, AI Magazine, volume 17, pages 37–54, Univ Calif Irvine, Dept Comp & Informat Sci, Irvine, Ca, 92717 Gte Labs Inc, Knowledge Discovery Databases Kdd Project, Tech Staff, Waltham, Ma, 02254. American Association for Artificial Intelligence.

[Grassl1999] Grassl, W. (1999). The reality of brands: Towards an ontology of marketing. American Journal of Economics and Sociology, 58(2):313–319.

[Gruber1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199–220.

[Guarino1995] Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. International Journal of Human and Computer Studies, 43:625–640.

[Jans *et al.*2008] Jans, M., Lybaert, N., and Vanhoof, K. (2008). Data mining as a methodology for internal fraud risk reduction. Journal of Information Systems, 32:1–29.

[Jasper and Uschold1999] Jasper, R. and Uschold, M. (1999). A framework for understanding and classifying ontology applications. In IJCAI 99 Ontology Workshop, pages 16–21.

[Jurisica *et al.*1999] Jurisica, I., Mylopoulos, J., and Yu, E. (1999). Ontologies for knowledge management: An information systems perspective. In for Information Sciences, A. S., editor, Proceedings of the Annual Conference of the American Society for Information Sciences (ASISâ€™99). American Society for Information Sciences.

[Kamakura *et al.*2003] Kamakura, W. A., Wedel, M., de Rosa, F., and Mazzon, J. A. (2003). Cross-selling through database marketing: a mixed data factor analyzer for data augmentation and prediction. International Journal of Research in Marketing, 20:45–65.

[Leary *et al.*2004] Leary, C. O., Rao, S., and Perry, C. (2004). Improving customer relationship management through database/internet marketing. European Journal of Marketing, 38(3/4):338–354.

[Linstone and Turoff2002] Linstone, H. A. and Turoff, M. (2002). The Delphi Method - Techniques and Applications. Murray Turoff and Harold A. Linstone.

[Mylopoulos *et al.*2004] Mylopoulos, J., Jurisica, I., and Yu, E. (2004). Ontologies for knowledge management. Knowledge and Information Systems, 3:380–401.

[Nedellec and Nazarenko2005] Nedellec, C. and Nazarenko, A. (2005). Ontologies and information extraction. In LIPN Internal Report. LIPN.

[Newell and level1982] Newell, A. and level, T. (1982). The knowledge level. Artificial Intelligence, 18:87–127.

[Nogueira *et al.*2007] Nogueira, B. M., Santos, T. R. A., and ZÃ¡rate, L. E. (2007). Comparison of classifiers efficiency on missing values recovering: Application in a marketing database with massive missing data. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007).

[Nou and McGuiness, 2003] Natalya F. Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University, 2003.

[O'Brien2002] O'Brien, R. (2002). Theory and Practice of Action Research, chapter An Overview of the Methodological Approach of Action Research. Universidade Federal da Paraiba.

[Ozimek2004] Ozimek, J. (2004). Case studies: The 2003 information management project awards. Journal of Database Marketing & Customer Strategy Management, 12(1):55.

[Payne and Frow2005] Payne, A. and Frow, P. (2005). A strategic framework for customer relationship management. Journal of Marketing, 69(4):167–176.

[Peppers and Rogers1999] Peppers, D. and Rogers, M. (1999). One To One future: building relationships one customer at a time. Doubleday.

[Phillips and Buchanan2001] Phillips, J. and Buchanan, B. G. (2001). Ontology-guided knowledge discovery in databases. In ACM, editor, International Conference On Knowledge Capture 1st international conference on Knowledge capture, pages 123–130. International Conference On Knowledge Capture.

[Rowe and Wright2001] Rowe, G. and Wright, G. (2001). Expert opinions in forecasting: The role of the delphi technique. Principles of Forecasting, pages 125–144. Kluwer Academic Publishers,.

[Siqueira *et al.*2001] Siqueira, S. W., Silva, D., Uchoa, E., Braz, H., and Melo, R. (2001). An architecture for database marketing systems. Lecture Notes in Computer Science, 2113.

[Smith *et al.*2008] Smith, M. K., Welty, C., and McGuinness, D. L. (2008). OWL Web Ontology Language Guide. W3C.

[Sowa2000] Sowa, J. F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co.

[Udrea *et al.*2007] Udrea, O., Getoor, L., and Miller, R. J. (2007). Leveraging data and structure in ontology integration. In of the 2007 ACM SIGMOD international conference on Management of data, P., editor, International Conference on Management of Data, pages 449–460.

[Uschold and Gruninger2004] Uschold, M. and Gruninger, M. (2004). Ontologies and semantics for seamless connectivity. SIGMOD Rec., 33(4):58–64.

[Uschold and King1995] Uschold, M. and King, M. (1995). Towards a metthodology for building ontologies. AIAI-TR Workshop on Basic Ontological Issues in Knowledge Sharing, 1.

[Vernette1997] Vernette, E. (1997). Evaluation de la validation predictive de la methode delphi-leader. In Congres International de l AFM, page 988 1010.

[Watada and Yamashiro2006] Watada, J. and Yamashiro, K. (2006). A data mining approach to consumer behavior. In Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06).

[Wehmeyer2005] Wehmeyer, K. (2005). Aligning it and marketing - the impact of database marketing and crm. Journal of Database Marketing & Customer Strategy Management, 12(2):243.

[Welty and Murdock2006] Welty, C. and Murdock, J. W. (2006). Towards knowledge acquisition from information extraction. In Springer, editor, In Proceedings of ISWC-2006., Athens.

[Welty1996] Welty, C. A. (1996). An Integrated Representation for Software Development and Discovery. PhD thesis, Rensselaer Polytechnic Institute.

[Witten and Frank2000] Witten, I. H. and Frank, E. (2000). Data Mining: Practical Machine Learning Tools and Technique. The Morgan Kaufmann Series in Data Management Systems, 2nd edition.

[Yin2003] Yin, R. K. (2003). Applications of case study research, volume 34 of Applied social research methods series. Sage Publications, Thousand Oaks, CA.

[Zairate *et al.*2006] Zairate, L. E., Nogueira, B. M., Santos, T. R. A., and Song, M. A. J. (2006). Techniques for missing value recovering in imbalanced databases: Application in a marketing database with massive missing data. In IEEE International Conference on Systems, Man, and Cybernetics, pages 2658–2664, Taiwan. IEEE.

[Zhou2007] Zhou, L. (2007). Ontology learning: state of the art and open issues. Information Technology and Management, 8:241–252.

[Zhou *et al.*2006] Zhou, X., Geller, J., and Halper, Y. P. M. (2006). An Application Intersection Marketing Ontology, chapter Theoretical Computer Science, pages 143–163. Lecture Notes in Computer Science. Springer Berlin / Heidelberg.

[Zubber-Skerritt2000] Zubber-Skerritt, O. (2000). New Directions in Action Research. Falmer Press.

[Zuber-Skerrit and Perry2000] Zuber-Skerrit and Perry, C. (2000). Action research in graduate management theses. Action Learning, Action Research and Process Management: Theory, Practice, Praxis, 1:84.

4.3 Ontological KDD Assistance

4.3.1 Introduction

This section describes a research of an ontological approach for leveraging the semantic content of ontologies to effectively support the knowledge discovery in databases. We analyze how ontologies and knowledge discovery process may interoperate and present our efforts to bridge the two fields, knowledge discovery in databases and ontology learning for successful database usage projects.

KDD is user dependent (Phillips and Buchanan, 2001). Thus, considering user interactions with process we use the ontologies to support such interactions.

There are different relevant topics to the KDD processes assistance also referred in literature such as "domain knowledge in KDD" (Domingos, 2003) (Kopanas *et al.*, 2002), "ontology/KDD integration" (Euler and Scholz, 2004) (Kuo *et al.*, 2007b)(Nigro *et al.*, 2008), "KDD life cycle"(Kotasek and Zendulka, 2000)(Diamantini *et al.*, 2004)(Cellini *et al.*, 2007) and "KDD assisted process" (Honavar *et al.*, 2001)(Bernstein *et al.*, 2005).

This section focuses on the ontologies role in the KDD process. It presents a hybrid process, ontology assisted KDD process, which leverages both ontology engineering and KDD taking in consideration the best industry and research practices. A brief application of assistance work in the KDD life cycle is depicted in Figure 13.

To accomplish this we have used the past KDD knowledge base in practical KDD supporting data analyst in each process phase during the complete process development. Moreover, using ontology and KDD process interoperation, we have also found a general knowledge base record dataset for updating purposes.

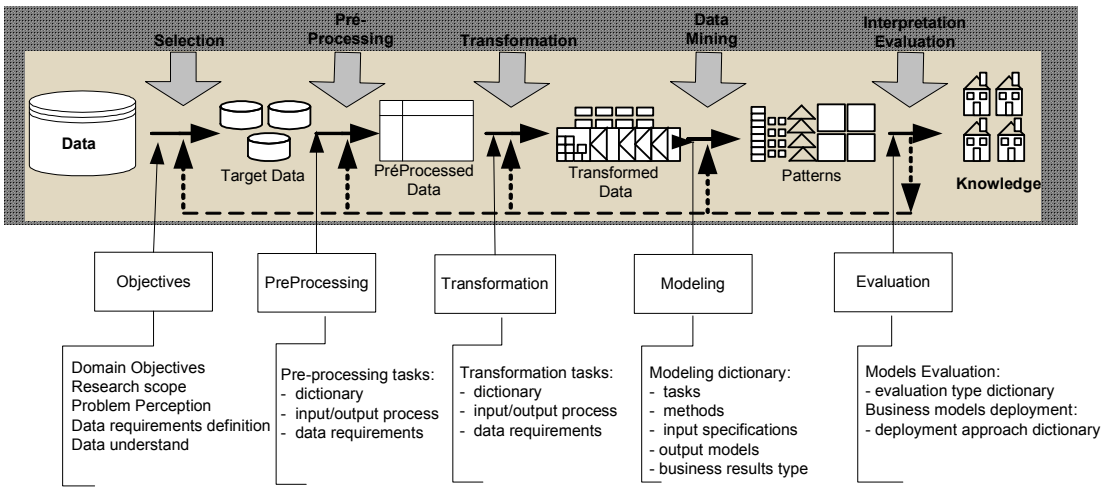


Figure 13: KDD general phase and task description workflow

For this we have used a real card loyalty program marketing database from an oil company, into a marketing objective: to discover the card owner (and user) profile use.

4.3.2 Research approach

Here, we attain to the effective support the KDD process using ontologies. In order to do this we have collected a marketing database from a multinational oil company. Therefore, we used a real case study in order to effectively test and deploy to ontological work in the KDD process. Our approach is defined in two distinct steps:

- Marketing database collection and preparation;
- Practical KDD ontology based development;

Marketing database

One of the most important marketing tools used by oil companies for customer fidelization is the marketing card loyalty programs. This approach allows cardholders to obtain fuel purchase discounts, to participate in marketing campaigns or to become members of a restrict club with restrict privileges.

Since it is an open marketing system program where all oil customers may access, from the company perspective this will turn into an important information source for almost customer oriented marketing strategies definition or product offer policies.

We have collected a card loyalty program marketing database from a multinational company. This database has three main tables: card owner, station and transactions. The available data refers to the past two year's activity. The data structure is as follows:

Table 4 - Data table card owner

Field	Description	Type	Domain range
IdCard	card identification	Primary key	
Idclient	Client identification	Foreigner key	
birthDate	Client birth date	Date	< today
cardClientDate	Starting card owner date	Date	<today
cardInitialDate	Starting client date	Date	<today
postcode	Zip code	integer	<10000
postCod3	3 Zip code	integer	<1000
maritalStatus	Marital status	String	{cas; sol; div; viv; out}
Gender	Client sex	String	{mas,fem}
vehicleType	Vehicle type description	string	{lig, merc, pes, out}
vehicleYear	Vehicle identification year date	number	<10000
fuelType	Fuel description type	string	{diesel, gasolina, gpl, out}

Table 5: Data table card transactions

Field	Description	Type	Domain range
IdMov	Transaction identification	Primary key	
IdCard	Card identification	Foreigner key	
date	Transaction date	Date	< today
fuelValue	Fuel transaction amount	real	<10000
fuelLitres	Transaction liters amount	real	<3000
shopValue	Transaction value amount	real	<10000
shopUnits	Shop units transaction amount	integer	<10
stationCode	Fuel station identification	Foreigner key	

Table 6: Data table station

Field	Description	Type	Domain range
stationCode	Fuel station identification code	Primary key	
stationType	Station identification type	String	{urb, rur, est}
postCode	Zip code	integer	<10000
postCod3	3 Zip code	integer	<1000

Practical KDD ontology based development

For the KDD development we have based our work on the free open source WEKA toolkit [Witten and Frank2000]. For ontological support we have used the PROTÉGÉ OWL editor and SWRL language [Knublauch2004].

Since KDD process generates output models, it was considered useful to represent them in a computable way. Such representation works as a general description of all options taken during the process. Based on PMML descriptive DM model we have introduced an OWL class in our ontology named *ResultModel* which holds instances with general form:
ResultModel { domain Objective Type; algorithm; algorithmTasks; algorithmParameters; workingAlgorithmDataSet; EvaluationValue; DeploymentValue}

Since the ontology contribution to the KDD process is quantitatively uncertain we have used a quality approach based on KDD team individual expert contribution.

Besides the above database, we also have used previous work results done with another database. The database belongs to a multinational distribution organization and contains all data related to a relationship marketing program, with more than 600 000 clients. The achieved results were validated and published in scientific publications (Santos *et al.*, 2005)(Pinto, 2006). Such work was developed during the author's master degree in technologies and information systems program.

4.3.3 Results

To build up mining experiments we have used Weka Toolkit [Witten and Frank2000] which allowed not only the actual mining but also featured analysis and algorithm evaluation. These experiments did not aim to the full construction of a classification model but instead to test and analyze different approaches and further ranking.

One of the algorithms use was performed throughout the following code, e.g., J48 parameters settings:

hasAlgoritm(alg)→J48
hasAlgorithmParameter (gainRatio) = workingDataSet(wds) → hasModel(?m)

The most relevant rule extracted from above data algorithms use was:

if (age<27 and vehicleType="Lig" and sex="Female") then 1stUsed="p"

In this model we may say that, female card owners under 27 years of age have a "*lig*" (ligeiro) category car and use a fuel station located in range of 10 kilometers from their address.

Also, practical KDD process tasks have been done supported by SWRL ontology queries. This query tasks was manually performed by the user. Therefore, the guidance was accomplished and achieved throughout knowledge base updating with the general model:

INSERT record KNOWLEDGE BASE
hasAlgorithm(J48) AND
hasModelingObjectiveType(classification) AND
hasAlgorithmWorkingData ({idCard; age; carClientGap; civilStatus; sex;
vehicleType; vehicleAge; nTransactions; tLiters; tAmountFuel;
tQtdShop; 1stUsed; 2stUsed; 3stUsed }) AND
Evaluation(67,41%; 95,5%) AND
hasResultMoldel (J48;classification; "wds",PCC;0,674;0955)

The evaluation, once performed, the system automatically updates the knowledge base with a new record. The registered information will serve for future use – knowledge sharing and reuse.

From the aforementioned previous research work we also have used the output models and integrated them into the knowledge base.

Table 7: distribution relatioship marketing database main attributes

Attribute	Domain
1. Household	{Non response, 1, 2, 3, 4, 5, 6 or more}
2. Dishwasher	{Non response, Yes, No}
3. Monthly Consumption (€)	{Non response, [0…150], [151…350], [351…500], [501…650], [651…[}
4. Household Income (€)	{Non response, [0…500],[501..750], [751…1000],[1001…1500], [1501…2250], [2251…[}
5. Children	{Non response, No, Yes}
6. Number of Children	{Non response, 1, 2, 3, 4, 5, 6, 7, 8, 9, [10…[}
7. Voucher Use	{No, Yes}

The objective was to determine the main customer profile regarding a marketing promotional discount voucher use. For this, several tasks were performed and have conducted to the following main results:

Rule 1
If (Dishwaher?)= "Yes" and
if(Children?)=" Yes" and
if(Household?)= 4 and
if (Monthly Consumption?) =" [151…350]" or "[501,750]"or " [750…[" then
VoucherUse="Yes"

Rule 2
If (Dishwasher?)= "Non Response" and
if(Children?)=" Yes" and
if(Household?)= 4 and
if (Monthly Consumption?)=" [151…350]" then
VoucherUse="Yes"

Therefore we have used above models in order to validate the effective KDD process assistance with the ontology. To do this, we have manually performed some SWRL

rules. This way, the guidance was accomplished and the knowledge base was updated with both models, such as:

INSERT record KNOWLEDGE BASE

hasAlgorithm(C5) AND hasAlgorithm(SOM)
hasModelingObjectiveType(classification) AND
hasAlgorithmWorkingData ({household; dishwasher; montlyConsumptionid; houseHoldIncome; children; numberChildren; voucherUse) AND
Evaluation(84,21%; 29,1%) AND
hasResultMoldel (C5; SOM; classification;"wds",PCC;0,84;0,29)

The complete prototype development and test in order to combine ontological engineering and KDD process is presented in the following section.

4.3.4 Ontological Assistance for Knowledge Discovery in Databases Process

Published in:

WSEAS Transactions on Information Science and Applications

World Scientific and Engineering Academy and Society

in press.

Abstract

The dramatically explosion of data and the growing number of different data sources are exposing researchers to a new challenge - how to acquire, maintain and share knowledge from large databases in the context of rapidly applied and evolving research. This paper describes a research of an ontological approach for leveraging the semantic content of ontologies to improve knowledge discovery in databases. We analyze how ontologies and knowledge discovery process may interoperate and present our efforts to bridge the two fields, knowledge discovery in databases and ontology learning for successful database usage projects.

1 Introduction

In artificial intelligence, ontology is defined as a specification of a conceptualization [Gruber1993]. Ontology specifies at a higher level, the classes of concepts that are relevant to the domain and the relations that exist between these classes. Indeed, ontology captures the intrinsic conceptual structure of a domain. For any given domain, its ontology forms the heart of the knowledge representation.

In spite of ontology-engineering tools development and maturity, ontology integration in knowledge discovery projects remains almost unrelated.

Knowledge Discovery in Databases (KDD) process is comprised of different phases, such as data selection, preparation, transformation or modeling. Each one of these

phases in the life cycle might benefit from an ontology-driven approach which leverages the semantic power of ontologies in order to fully improve the entire process [Gottgtroy *et al.*2004].

Our challenge is to combine ontological engineering and KDD process in order to improve it. One of the promising interests in use of ontologies in KDD assistance is their use for guiding the process. This research objective seems to be much more realistic now that semantic web advances have given rise to common standards and technologies for expressing and sharing ontologies [Bernstein *et al.*2005].

The three main operations of KDD can take advantage of domain knowledge embedded in ontologies such as: At the data understanding and data preparation phases, ontologies can facilitate the integration of heterogeneous data and guide the selection of relevant data to be mined, regarding domain objectives; During the modeling phase, domain knowledge allows the specification of constraints (e.g., parameters settings) for guiding data mining algorithms by, (e.g. narrowing the search space); finally, to the interpretation and evaluation phase, domain knowledge helps experts to visualize and validate extracted units.

KDD process is usually performed by experts who use their own knowledge for selecting the most relevant data in order to achieve domain objectives [Gottgtroy *et al.*2003]. Here we explore how the one ontology and its associated knowledge base can assist the expert at KDD process. Therefore, this document describes a research on a new approach to leveraging the semantic content of ontologies to improve KDD.

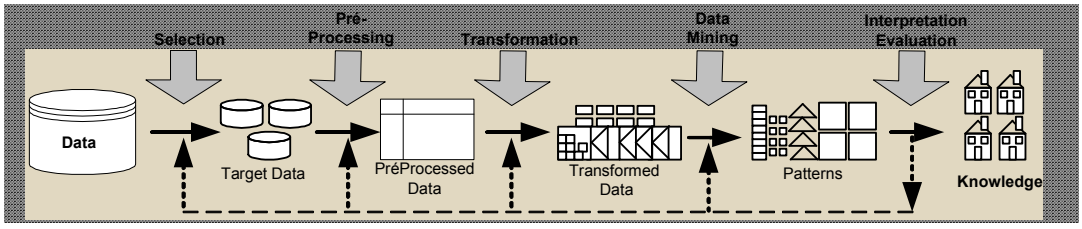


Figure 1: Knowledge discovery process framework adapted from [Fayyad *et al.*1996]

This paper is organized as follows: after this introductory part we present related background concepts. Then, we present related work on this area following the presentation and discussion of ontological assistance. The main contribution is presented in terms of a system prototype description and also system operation sections. Finally we draw some conclusions and address further research based on this research to future KDD data environment projects.

2. Background

2.1 Knowledge Discovery in Databases

Knowledge discovery in databases (KDD) is the result of an exploratory process in order to achieve domain defined objectives involving the application of various algorithmic procedures for manipulating data, building models from data, and manipulating the models. The Data Mining phase deserves more attention from the research community: processes comprise multiple algorithmic components, which interact in nontrivial ways.

We consider tools that will help data analysts to navigate the space of KDD processes systematically, and more effectively. In particular, this paper focuses on a subset of stages of the KDD —those stages for which there are multiple algorithm components that can apply.

For most of this paper, we consider a prototypical KDD process template, similar to the one represented in Figure 1.The sequence of KDD phases is not strict. Moving back and forth between different phases is always required. It depends on the outcome of each phase, which one, or which particular task of a phase has to be performed next.

We focus our attention on the three main macro components of KDD life cycle: data understanding (data selection); data pre processing (all related data preparation and transformation activities), and modeling (data mining and the application of induction algorithms) We have chosen this set of components because, individually, they are relatively well understood—and they can be applied to a wide variety of benchmark data sets.

2.2 Ontology Web Language

Ontologies are used to capture knowledge about some domain of interest. Ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. Ontology Web Language (OWL) is a standard ontology language from the World Wide Web Consortium (W3C).

An OWL ontology consists of: Individuals (represent domain objects); Properties (binary relations on individuals - i.e. properties link two individuals together); and Classes (interpreted as sets that contain individuals).

Moreover, OWL enables the inclusion of some expressions to represent logical formulas in Semantic Web rule language (SWRL) [Horrocks *et al.*2004]. SWRL is a rule language that combines OWL with the rule markup language providing a rule language compatible with OWL.

2.3. Semantic Web Language Rule

To the best of our knowledge there are no standard OWL-based query languages. Several RDF -based query languages exist but they do not capture the full semantic richness of OWL. To tackle this problem, it was developed a set of built-in libraries for Semantic Web Rule Language (SWRL) that allow it to be used as a query language

The OWL is a very useful means for capturing the basic classes and properties relevant to a domain. However, these domain ontologies establish a language of discourse for eliciting more complex domain knowledge from subject specialists. Due to the nature of OWL, these more complex knowledge structures are either not easily represented in OWL or, in many cases, are not representable in OWL at all. The classic example of such a case is the relationship *uncleOf(X,Y)*. This relation, and many others like it, requires the ability to constrain the value of a property (*brotherOf*) of one term (*X*) to be

the value of a property (*childOf*) of the other term (*Y*); in other words, the *siblingOf* property applied to *X* (i.e., *brotherOf(X,Z)*) must produce a result *Z* that is also a value of the *childOf* property when applied to *Y* (i.e., *childOf(Y,Z)*). This "joining" of relations is outside of the representation power of OWL.

One way to represent knowledge requiring joins of this sort is through the use of the implication (→) and conjunction (AND) operators found in rule-based languages (e.g., SWRL). The rule for the uncleOf relationship appears as follows:

$$brotherOf(X,Z) AND\ childOf(Y,Z) \rightarrow uncleOf(X,Y)$$

2.4 Evaluation of knowledge reuse effectiveness

The main objective of this research is to assist the KDD process based on ontology knowledge. Therefore, it is assumed that effectively ontology has learned from KDD domain and practice. Thus it is possible to provide users with information that are relevant to their needs at each of KDD phases.

Hence, the related task (process option) suggestion returned by the ontology will be the primary basis to determine the quality of the relevant information retrieved. For the present purpose we admit as major indices, precision and recall [Han and Kamber2001]:

$$Precision=(Relevant \cap Selected)/(Selected\_Results)$$

Precision expresses the proportion of related results (Relevant∩Selected) among relevant results retrieved (Selected_Results). In other words, to reflect the amount of knowledge correctly identified (in the ontology) with respect to the whole knowledge available in the ontology As related results we intend the entire set of ontology elements (classes and data properties) related to the subject (e.g., to preprocessing phase: set of related classes and relationships). Also, we use selected results as the set of related results and selected at the user question (e.g., set of suggested results for, e.g., birthDate attribute preprocessing).

Recall=(Relevant ∩Selected)/(Relevant_Results)

Recall expresses the proportion of results retrieved (Relevant∩Selected) from related results (Relevant_Results). It is used to reflect the amount of knowledge correctly identified with respect to all the knowledge that it should identified.

In our work, precision will be used to evaluate the proportion of user interests towards the KDD phase assistance. This proportion examines how correct the ontology is suggesting tasks (options) when solicited by the user. On other hand, recall, estimates the ability that the system is able to satisfy user needs.

Since there is an inverse impact between precision and recall measures, we combine both indices through a Precision Recall Index (PRI) computation. This index estimates the ontology output recommendation to avoid the condition of inverse impact between precision and recall. The expression used is as follows [Sarwar *et al.*2000]:

PRI=(2*Precision*Recall)/(Precision+Recall)

We have introduced these indexes in order to provide the user with further information about the option they need, e.g., to the *birthDate* attribute the user will have an answer like:

birthDate: preprocessing (precision;recall;pri);
birthDate: AttributeDerivation (precision;recall;pri);
birthDate: outliers (precision;recall;pri);

3. Related work

A KDD assistance through ontologies should provide user with nontrivial, personalized "catalogs" of valid KDD-processes, tailored to their task at hand, and helps them to choose among these processes in order to analyze their data.

In spite of the increase investigation in the integration of domain knowledge, by means of ontologies and KDD, most approaches focus mainly in the DM phase of the KDD process [Anand *et al.*2007] [Bernstein *et al.*2005] [Domingos2003] while apparently the role of ontologies in other phases of the KDD has been relegated.

Currently there are others approaches being investigated in the ontology and KDD integration, like ONTO4KDD or AXIS . Both of them are focusing the application of ontologies in order to improve overall KDD process regarding DM models optimization and sophistication.

In the literature there are several knowledge discovery life cycles, mostly reflect the background of their proponent's community, such as database, artificial intelligence, decision support, or information systems [Gomez-Perez *et al.*2004]. Although scientific community is addressing ontologies and KDD improvement, at the best of our knowledge, there isn't at the moment any fully successful integration of them.

This research encompasses an overall perspective, from business to knowledge acquisition and evaluation. To this end we use a DM ontology (DMO), integrated in KDD process to propose a general framework. Moreover, this research focuses the KDD process regarding the best fit modeling strategy selection supported by ontology.

Therefore, at this research we focus the role of ontology in order to assist the KDD in different process stages': data understand; data preparation and modeling. Indeed, to select the appropriate an adequate tasks sequence to support the KDD work becomes an important decision. This work proposes a computational model based on ontologies to assist the KDD planning process.

4. Ontological work

This research work is a part of one much larger project: Database Marketing Intelligence supported. by ontologies and knowledge discovery in databases. Since this

research paper focuses the KDD process ontological assistance, we mainly focus this research domain area.

Most of ontology building methodological approaches reported are mainly overall lifecycle. They provide a more generic framework for the ontology creation process, but giving little support for the actual task of building the ontology. To develop our data preparation phases ontology we have used the METHONTOLOGY methodology [Gomez-Perez *et al.*2004][Fernandez *et al.*1997][Blazquez *et al.*1998]. This methodology best fits our project approach, since it proposes an evolving prototyping life cycle composed of development oriented activities:

requirements specification: through conceptualization of domain knowledge, formalization of the conceptual model in a formal language and implementation of the formal model;

support oriented activities: focuses knowledge acquisition, the ontology documentation, evaluation and if the case integration of other ontologies;

project exploration and management activities: concerns all related ontology use and further maintenance.

Since this has been done elsewhere, the work related in this paper focuses only the ontology use at the KDD process. It will depict the development oriented activities within the above methodology and provide a more specific methodology for this part.

The methodology presented here focuses on the actual acquisition and development part of the ontology and describes a comprehensive, reusable and semi automatically-supported framework, which can be embedded in other KDD lifecycle models.

4.1. Ontology construction

Through an exhaustive literature review we have achieve a set of domain concepts and relations between them to describe KDD process. Following METHONTOLOGY we had constructed our ontology in terms of process assistance role. Nevertheless, domain concepts and relations were introduced according some literature directives [Blazquez *et al.*1998][Smith and Farquhar2008].

Moreover, in order to formalize all related knowledge we have used some relevant scientific KDD [Quinlan1986] [Fayyad *et al.*1996, Fayyad and Uthurusamy1996] [Agrawal *et al.*1993] and ontologies [Phillips and Buchanan2001] [Nigro *et al.*2008] published works. However, whenever some vocabulary is missing it is possible to develop a research method (e.g., through Delphi method [Delbecq *et al.*1975] [Chu and Hwang2008] [Pinto *et al.*2009a] [Pinto *et al.*2009b]) in order to achieve such a domain knowledge thesaurus.

At the end of the first step of methontology methodology we have identified the following main classes (Figure 2):

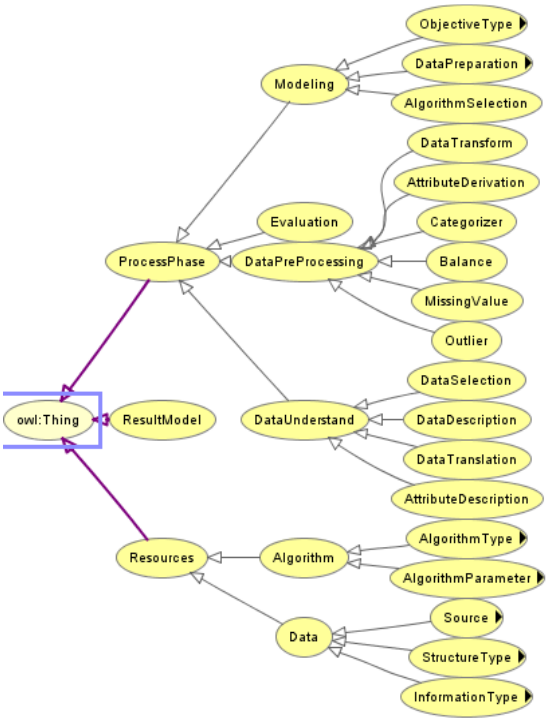


Figure 2: KDD ontology class taxonomy (partial view)

Our KDD ontology has three major classes: *Resource*, *ProcessPhase* and *ResultModel*. ProcessPhase is the central class which uses resources (Resource class) and has some results (ResultModel class). The former Resource class relates all resources needed to carry the extraction process, namely algorithms and data. The ResultModel has in

charge to relate all KDD instance process describing all resources used, all tasks performed and results achieved in terms of model evaluation and domain evaluation.

Regarding KDD process we have considered four main concepts below the ProcessPhase concept (OWL class):

Data Understand focuses all data understanding work from simple acknowledge attribute mean to exhaustive attribute data description or even translation, to more natural language;

Data Preprocessing: concerns all data pre-processing tasks like data transformation, new attribute derivation or missing values processing;

Modeling: Modeling phase has in charge to produce models. It is frequent to appear as data mining phase (DM), since it is the most well known KDD phase. Discovery systems produce models that are valuable for prediction or description, but also they produce models that have been stated in some declarative format, that can be communicated clearly and precisely in order to become useful. Modeling holds all DM work from KDD process. Here we consider all subjects regarding the DM tasks, e.g., algorithm selection or concerns relations between algorithm and data used (data selection). In order to optimize efforts we have introduced some tested concepts from other data mining ontology (DMO) [Nigro *et al.*2008], which has similar knowledge base taxonomy. Here we take advantage of an explicit ontology of data mining and standards using the OWL concepts to describe an abstract semantic service for DM and its main operations. Settings are built through enumeration of algorithm properties and characterization of their input parameters. Based on the concrete Java interfaces, as presented in the Weka software API [Witten and Frank2000] and Protégé OWL, it was constructed a set of OWL classes and their instances that handle input parameters of the algorithms. All these concepts are not strictly separated but are rather used in conjunction forming a consistent ontology;

Evaluation and Deployment phase refers all concepts and operations (relations) performed to evaluate resulting DM model and KDD knowledge respectively.

Then, we have represented above concept hierarchy in OWL language, using protégé OWL software.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.semanticweb.org/ontologies/2009/5/DBMiPhDfpinto.owl">
 <owl:Class rdf:ID="InformationType">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Data"/>
  </rdfs:subClassOf>
 <owl:Class rdf:ID="Personal">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="lnformationType"/>
  </rdfs:subClassOf>
 </owl:Class>
 </owl:Class>
 <owl:Class rdf:ID="Demographics">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Personal"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
 <owl:Class rdf:about="#InformationType">
  <rdfs:subClassOf rdf:resource="#Data"/>
 </owl:Class>
```

Following Methontology, the next step is to create domain-specific core ontology, focusing knowledge acquisition. To this end we had performed some data processing tasks, data mining operations and also performed some models evaluations.

Each class belongs to a hierarchy (Figure 3). Moreover, each class may have relations between other classes (e.g., PersonalType is-a InformationType subclass). In order to formalize such schema we have defined OWL properties in regarding class' relationships, generally represented as:

Modeling^ has Algorithm(algorithm)

In OWL code:

```
 <owl:Class rdf:ID="AlgorithmSelection">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:someValuesFrom rdf:resource="#Algorithms"/>
    <owl:onProperty>
```

```
      <owl:ObjectProperty rdf:ID="hasAlgorithm"/>
     </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
   <owl:Class rdf:ID="Modeling"/>
  </rdfs:subClassOf>
 </owl:Class>
```

The ontology knowledge acquisition, firstly, happens through direct classes, relationships and instances load. Then through the KDD instantiation, the ontology acts according to the semantic structure.

Each new attribute is presented to the ontology, it is evaluated in terms of attribute class hierarchy, and related properties that acts according it.

In our ontology Attribute is defined by a set of three descriptive items: *Information Type*, *Structure Type* and allocated *Source*. Therefore it is possible to infer that, Attribute is a subclass of *Thing* and is described as a union of *InformationType*, *StructureType* and *Source*.

At other level, considering that, data property links a class to another class (subclass) or links a class with an individual, we have in our ontology the example:

StructureType(Date)
→ *hasMissingValueTask*
→ *hasOutliersTask*
→ *hasAttributeDerive*

Attribute InformationType (Personal) & Attribute PersonalType(Demographics)
→ hasCheckConsistency

As example, considering the *birthDate* attribute, ontology will act as:

? Attribute hasDataSource
attribute hasDataSource (CustomerTable).
? Attribute hasInformationType:

Attribute hasInformationType (Personal) then:
attribute hasPersonalType(Demographics)
? Attribute hasStructureType
attribute hasStructureType (Date).
: attribute hasStructureType(Date) AND
PersonalType(Demographics) then:
: attribute (Demographics; Date) hasDataPreparation
: attribute (Demographics; Date) hasDataPreProcessing
AND Check missing values
AND Check outliers
AND Check consistency
AND deriveNewAttribute

In above example, the inference process is executed on reasoner for description logic (Pellet). It acts along both class hierarchy (e.g., *Personal or Demographics*) and defined data properties (e.g., *hasStructureType* or *hasDataPreparation*). In above example the attribute belongs at two classes: *Date* and *Demographics*. Through class membership, the *birthDate*, attribute inherits related data properties, such as *hasDataPreparation* or *hasDataPre-Processing*

4.2 Ontology Leaning cycle

Ontology assistance to KDD aims the improvement of the process allowing both better performance and extracted knowledge results.

Since KDD process is the core competency of database use, it is the centre focus of our work.

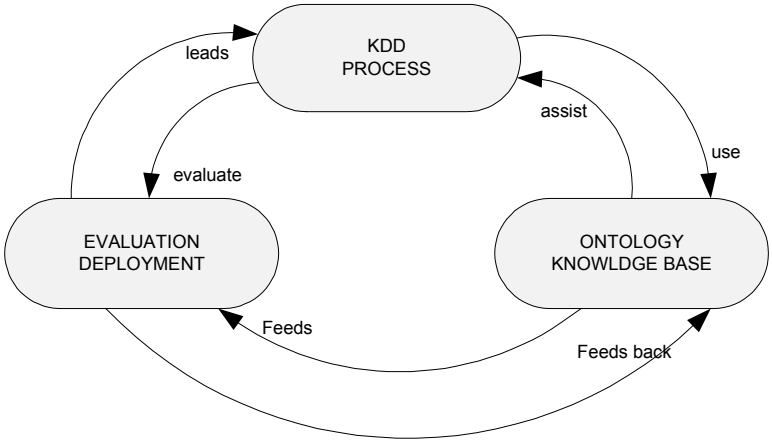


Figure 3: Ontology learning cycle

As depicted in Figure 3, KDD process is located at the centre of our system. Therefore, data analyst uses knowledge during the process execution; knowledge feeds performance for higher achievement, and performance leads measures performance through evaluation and deployment methods; performance feeds back knowledge (ontology update) for later use of that knowledge. Also knowledge drives the process to improve further operations.

Since the KDD process generates as output models, it was considered useful to represent them in a computable way. Such representation works as a general description of all options taken during the process. Based on PMML descriptive DM model we have introduced an OWL class in our ontology named *ResultModel* which holds instances with general form:

ResultModel {
domain Objective Type;
algorithm;
algorithmTasks;
algorithmParameters;
workingAlgorithmDataSet;
EvaluationValue;
DeploymentValue
}

Moreover, our ontology has the learning capability mutually assigned to aforementioned model the ontology structure (precision and recall index). Then it is possible both: so suggest (e.g., algorithm) and rank each suggestion (e.g., accuracy). Such approach may lead in a future to the development of an automatic learning capability

4.3 Knowledge Reuse

One of the promising interest of ontologies is they common understand for sharing and reuse. Hence we have explored this characteristic to effectively assist the KDD process. Indeed, this research presented the KDD assistance at two levels:

- Overall process assistance based on *ModelResult* class. Since this class is used as previous KDD process repository, the system use the ontology to infer accordingly some defined inputs, e.g., *swrl:query modelresult* (*?do,"classification"*);
- KDD phase assistance. Since our ontology has a formal structure related to KDD process, is able to infer some result at each phase. To this end, user need to invoke the system rule engine (reasoner) indicating some relevant information, e.g., at data preprocessing task: *swrl:query hasDataPreprocessingTask(?dpp,"ds")*, where *hasDataPreProcessingTask* is an OWL property which infers from ontology all assigned data type preprocessing tasks (*dpp*) related to each attribute type within the data set "*ds*". Moreover, user is also assisted in terms of ontology capability index, through the ontology index - precision, recall and PRI metrics.

Once we have a set of running KDD process registered at the knowledge base, whenever a new KDD process starts one the ontology may support the user at different KDD phases. As example to a new classification process execution the user interaction with ontology will follow the framework as depicted in Figure 4:

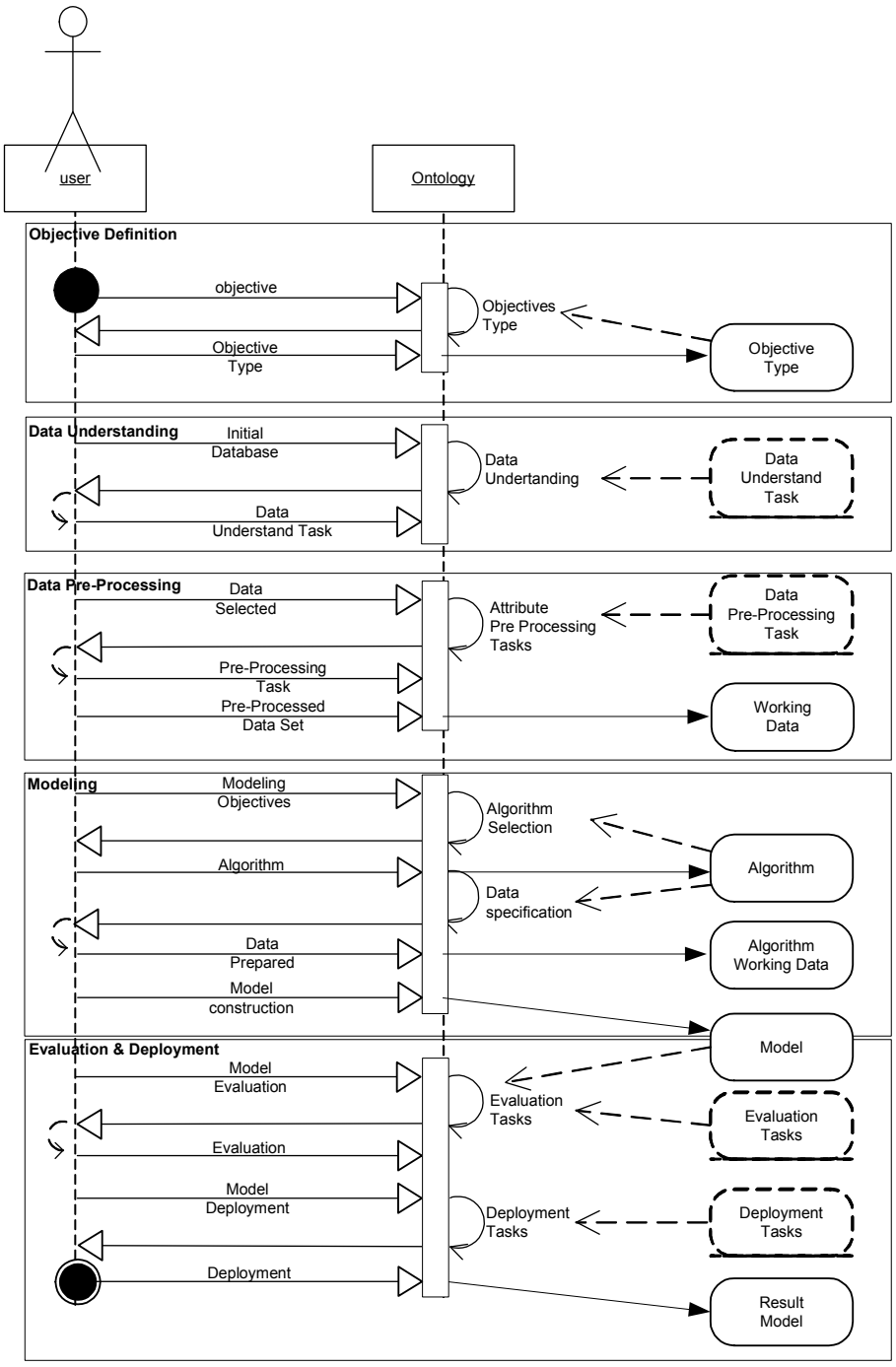


Figure 4: KDD ontological assistance sequence diagram

The ontology will lead user efforts towards the knowledge extraction suggesting by context. That is the ontology will act accordingly to user question, e.g., at domain objective definition (presented by user) the ontology will infer which is type of objectives does the ontology has. All inference work is dependent of previous loaded

knowledge. Hence, there is an ontology limitation – only may assist in KDD process which has some similar characteristics to others already registered.

5. System Prototype

A general overview of the main components of the system is shown in Figure 5. Our system has four main components, as depicted in Figure 5:

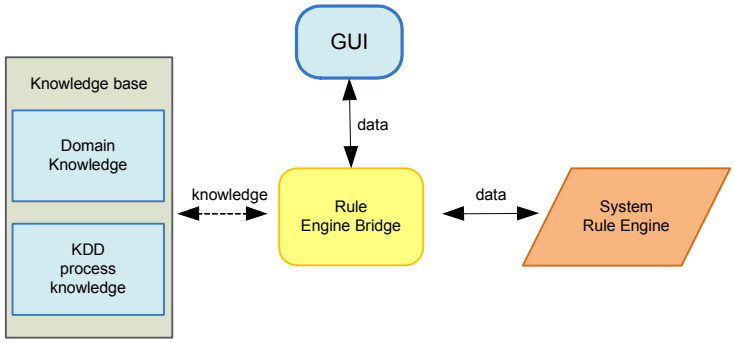


Figure 5: Ontology assistance system components

- **Knowledge base:** developed over Protégé OWL editor is used to create and maintain the ontology. Protégé stores information the OWL format file. The knowledge base is formed by two main components: domain knowledge ontology and KDD process ontology – here, for modeling purposes, we have introduced some ontology concepts from Data Mining Ontology [Nigro *et al.*2008];
- **Rule engine bridge:** performs inference tasks through OWL knowledge base. It extracts SWRL rules and relevant OWL knowledge, using the rule engine and system knowledge base. To infer about knowledge in the knowledge base, we build SWRL expressions to perform queries over the knowledge base and invoke the Pellet reasoner [Parsia and Sirin2004]. We need implement engine or map to the existing rule engine, here the bridge;
- **System rule engine:** is based on SWRL API supported by Jena Toolkit [McBride2002] and is able to interact with a user to assemble the required information. Jena is a Java framework for building Semantic Web applications. It provides a programming environment for RDF, RDFS, OWL and SPARQL

and includes a rule based inference engine. Jena is available to Protégé through an API – JessTab [Friedman-Hill and Scuse2008];

- **GUI:** developed through Eclipse java software to develop it supports the system user interface.

Keeping it straight forward, the assistant system communicates over the rule engine bridge with the Pellet reasoner, which is able to answer a subset of SWRL/SPARQL queries [Seaborne2004]. Also the inference system queries knowledge base every time it needs to enumerate some parameters or find a DM task, algorithm, service, and so forth. Moreover, our system also updates the knowledge base with instances of DMO classes and values of their properties.

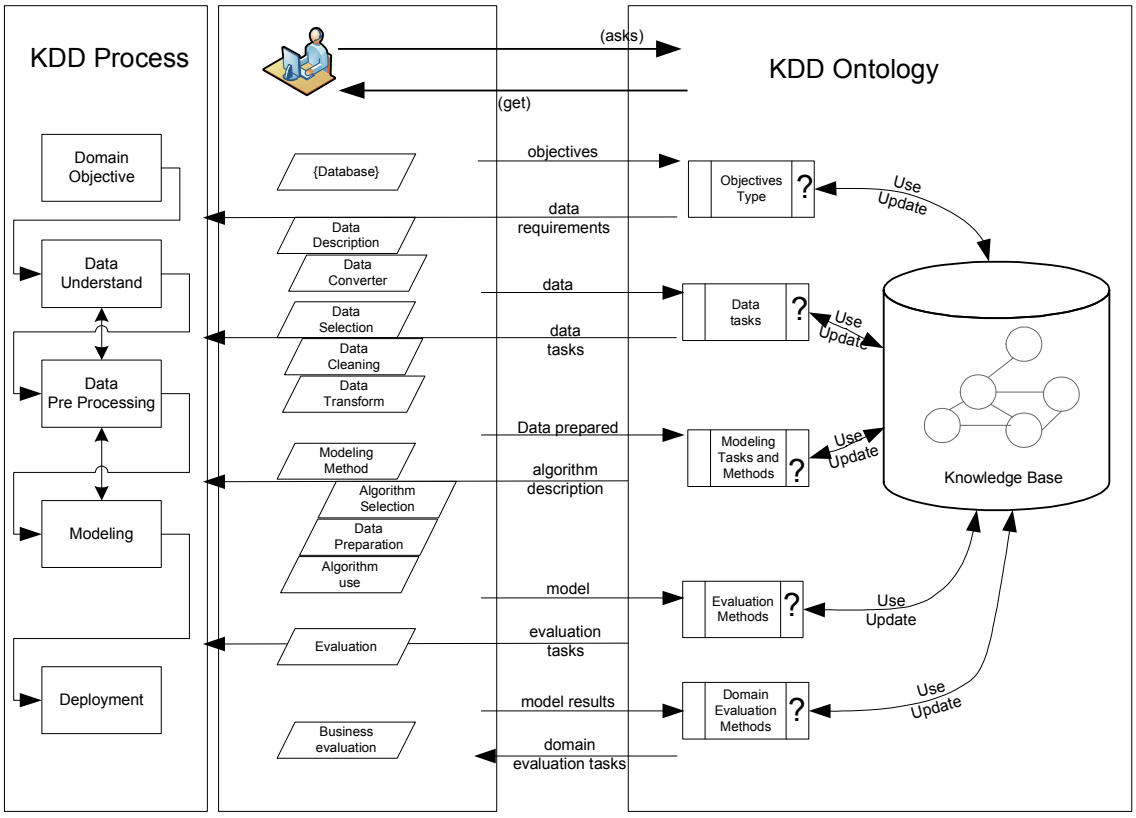


Figure 6: Ontology assistance to knowledge discovery process

6. Ontological Assistance

To achieve the goals presented in the introductory section, we have designed a specialized tool that fulfils the role of the KDD shown in Figure 1. For simplicity reasons, trough Eclipse software it is implemented an application able to navigate a user in the KDD process.

We attain to assist the user to carry out the KDD process from the domain objective until extracted knowledge evaluation. Indeed, our solution provides a support to choose particular knowledge extraction objectives and manage the entire process, from data to output models evaluation.

Our proposal for KDD assistance has three main reference layers (Figure 6): KDD running process phases; user interface; and knowledge base support layer. Each one of these layers follow a general process framework orientation, that is, since our objective it to support and assist the KDD process, all ontological work is done accordingly with this referential.

i. At a user perspective our assistance framework begins at the early domain objective definition. Each domain objective may have a more general objective which may be useful to the rest of the process, e.g., in relationship marketing, we may have three kinds of objectives: customer identification, fidelization, personalization and customization;
ii. At data understand phase, the user acts with ontology supplying the database (set of attributes and records) and as result receives a data task list accordingly, that is, ontology will infer about attribute data type and quality suggesting a set of tasks, e.g., to attribute with date type, it will be recommended to check the data consistency - client active birth date attribute must be older than today and earlier than a reference date (*1-1-1900*);
iii. Data preprocessing: since it is recognized as one of the most KDD tasks time consuming phase, the ontology play a significant role. It may be use overwhelm many user limitations in terms of data preprocessing tasks perspectives. Much of the KDD success depends on user insights over the data. Then, considering the ontology as repository the user may get a useful data preprocessing task list, as

example only one attribute (e.g., *transactionDate*) may be derived into many more others, like, e.g., *firstTransactionDate*, *lastTransactionDate, meanTimeBetween-Transaction; transactionsPerPeriod* and so on;

iv. At modeling level our user need to indicate which are they modeling objectives. Then ontology will infer throughout his knowledge base, in terms of available attributes, data characteristics and model requirements. Our ontology KDD assistant determines which modeling approach is more appropriate. As example, a fidelity card marketing database where the domain objective is the customer profile. A decision-tree (e.g. C5.0) learning alone might be appropriate. Or, a decision-tree learner plus sub-sampling as pre-process, or plus pruning as a post-process, or plus both. Other options might be: are Naıve Bayes or Self Organizing Maps neural networks also appropriate? Perhaps not by themselves. Not so, if the Naive Bayes implementation accepts only categorical attributes. On the other hand, neural networks often accept only numeric attributes. However, pre-processing to transform the attribute type may enable their use. Such wide spectrum of answers are available to the user, by the algorithms description ontology answer;

v. At ending KDD phase, model evaluation and deployment the ontology assists the user with models evaluation methods available (e.g., area under the curve or confusion matrix methods) and also mode appropriate domain methods to model deployment (e.g., customer set control group definition).

As presented in along previous sections, our system is designed to suggest best fit tasks at each KDD phase according to the knowledge base and user requirements. Moreover, the system dynamically modifies the task set composition depending on knowledge extraction objectives, entered data, defined preconditions and effects, and existing description of services available in the knowledge base. It corresponds to one of the most important KDD definitions “interactive and non trivial process”[Fayyad *et al.*1996]. As example of such capability, the following expression, demonstrates how each phase may be connect through an inference instruction:

getModelTask [(hasDomainObjective(?do)^
hasModelingObjective(?mo)
hasAlgorithmselection(?alg)^
hasDataSelection(?ds)]Model (?m)

This executable SWRL *getModelTask* expression is used to perform a set of modeling tasks accordingly with some KDD requirements, as domain defined objective (*hasDomainObjective*); modeling objective type (*hasModelingObjective*); algorithm use, through the algorithm selection (*hasAlgorithmSelection*); and data set to be used by selected algorithm (*hasDataSelection*). Each of these OWL properties invoke the knowledge base through the inference process, in order to achieve own specification

7. Experiment

Our system prototype operation follows general KDD framework (Figure 1) and uses the ontology to assist at each user interaction. Our experimentation was developed over a real oil company fidelity card marketing database. This database has the fowling structure and attributes:

Table: card owner

IdCard: card identification (pk) (number);
IdClient: cliente identification (fk)(number);
birthDate: client birth date (date);
cardInitialDate: issued card date (date);
clientInitialDate: starting client date (date);
postCode: main client postal code (number);
postCod3: three digit specification post code (number);
civilStatus: married status (set);
gender: client sex (set);
vehicleType: vehicle type description (set);
vehicleYear: vehicle production date (number);
fuelType: fuel description type (set).

Table: card transactions

idMov: transaction identification (pk) (number);
IdCard: card identification (fk) (number);
Date: transaction date (date);

fuelValue: amount transaction fuel value (real);
fuelLitres: amount transaction liters (number);
shopValue: amount transaction shop value (real);
shopUnits: amount transaction shop units bought (number);
stationcode: fuel station identification (fk) (number)

Table: station

stationCode: fuel station identification code (pk) (number);
stationType: station identification type (set);
postCode: main fuel station postal code (number);
postCod3: three digit specification post code (number);

To carry out this we have developed an initial set of SWRL rules. Since KDD is an interactive process, these rules deal at both levels: user and ontological levels. The logic captured by these rules is this section using an abstract SWRL representation, in which variables are prefaced with question marks.

Domain objective: customer profile
Modeling objective: description
Initial database: fuel fidelity card;
Database structure: 4 tables;

Attribute List:
customerTable {
Idcard; idclient; birthDate; cardInitialDate; clientInitialDate; postCode; postCod3; civilStatus; gender; vehicleType; vehicleYear; fuelType }
TransactionTable {
idMov; idCard, date, fuelValue, fuelLitres; shopValue; shoppUnits; stationcode}
StationTable{
stationCode; stationType; postCode}

Using some pseudo code based on SWRL, we take a closer look at ontological KDD assistance development process.

7.1. Objectives definition

At user level, our system uses the ontology to assist at objective type selection. This task is performed throughout the following SWRL code:

DomainObjective(?obj)-> query:user input

hasDomainObjectiveType(?do)

The user is prompted to select one of the already available objectives types - *query:user input hasDomainObjectiveType(?do)*. As result do variable will hold a domain objective type value, e.g.,"*classification*"

7.2. Data understanding and data selection

Data understanding stands for user data description, comprehension, and evaluation. Besides the domain knowledge required to understand the data, and prior use at KDD process, each attribute need to be evaluated by a set of analysis tasks, e.g., data completeness (missing values); data description (e.g., range values, units, granularity), among others.

Select Attribute (?att)
Identify Attribute Information Type (?att,?it)
Identify Attribute StructureType (?att,?st)
Data set description (numbers):

	Records	Attributes
Customer Table (original)	9285	13
Transaction Table	292427	9
Station Table	212	3
Working dataTable	9285	30

Initial data working set selection is carried through an individual attribute evaluation in terms of OWL data properties, as following example:

- *hasAttributeStructureType (?att)*: performs an identification of attribute data format, e.g., uniform value type;
- *hasAttributeInformationType(?att)*: evaluates the attribute in terms of standard information type;
- *hasMissingValue(?att)*: performs a data completeness evaluation in terms of e.g., missing values;

As a running example we may use the attribute *birthDate* to perform such attribute characteristics evaluation:

```
query:user input hasAttribute (birthDate;?att)
: hasAttributeInformationType (?att) → Personal;
   ::hasPersonalInformationtype(?att)→ Demographics
: hasAttributeStructureType(?att)→ Date
: hasMissingValue(?att) -> 0,05 { uncompleted records rate}
```

Firstly begins with attribute selection, prompting user with attribute list. Then, invokes some properties regarding attribute information type and attribute structure type. At the end we have used the properties and specialize inside personal class
This attribute will be assigned a record as:

```
{
        : Information Type : #Personal
        : Personal information Type: #Demographics
        : Structure Type: #Date
        : Missing Value :#5%
}
```

Data selection task (to form the working data set :*wds*) is therefore performed with according the previous data understand attribute record. As example:

```
Select wds from database
Where
        att.MissingValue<10% and (att.DataInformationType = "Personal" OR
        att.InformationType ="Transaction")
```

7.3. Data Preprocessing

Since we have selected working data (*wds*) it is needed to proceed with its preparation regarding algorithm's data format requirements. Therefore, previous any data pre-processing task it must be selected the modeling objective:

```
: domainObjective (?do) ^
  hasModelingObjectives(?do,?mo) modelingObjective(mo)
```

DataPreProcessingTask is a data property that selects and displays the data preprocessing task to be performed over the working data (*wds*).

```
: hasModelingObjective(?mo)^ hasWorkingData(?wds) ->sqwrl:select DataPreProcessingTask(?dpp)
```

Data pre-processing evolves a wide range of data tasks, like new attribute derivation, data normalization, data categorization, data reduction or data transformation.

```
:attributeInformationType (e.g., Personal) hasPre-ProcessingTask (list)
…
:attributeStructureType (e.g., Date) hasPre-ProcessingTask (list)
…
: attributeSourceType (e.g., externalDB) hasPre-ProcessingTask (list)
```

Getting back to our example, with *birthDate* attribute we will have the following code:

```
: attribute (?att; Personal; Date) hasDataPre-ProcessingTask
hasOutliers(?att,#validDateRange)
hasConsistency (?att, #validRule)
hasNewAttributeDerive(?att,#newAtt)
```

As result we will have:

- To outliers treatment a valid range is defined through *#validDateRange* – any value outside this range is marked as outlier;
- To consistency it is required a valid rule in order to evaluate if the record value is correct, e.g., *birthDate* must be older than *cardInitialDate* value;
- New attribute derivation is one of the most important pre-processing tasks, since this operation may provide the analyst with some useful new attributes – from birth date we may have others attributes like *age* or *horoscope sign*.

7.4 Modeling

The modeling phase objective is to mine data using previously selected and prepared data set throughout algorithms modeling. Such work may vary from single algorithm

use (direct use of e.g., apriori algorithm) or some complex algorithms use (e.g., self organizing maps neural networks in conjunction with J48 decision tree).

At modeling phase there exists several interactions looking for the best algorithm combination (attribute selection e.g., through random sampling; attribute categorization, parameters settings definition and finally, algorithm application) towards the best model performance.

For question of ontology demonstration simplicity we only focus the algorithm use and its performance results. The algorithm selection is carried though the following SWRL statements:

:WorkingDataSet(?ds)^ hasModelingObjectiveType(?mo)^ hasModelSelection(?wds,?mo)^
hasAlgorithmClass(?alg,?mo) → hasAlgorithm (?alg)

hasAlgorithm presents all algorithm options to the above specifications. Once algorithm is selected, it is necessary to fulfill some specifications (e.g., *hasAlgorithmParameter*), in order to achieve the algorithm output model (*?m*):

has Algorithm(?alg)^
hasAlgorithmParameter(?alg,?pSet)^
workingData(wds) → hasModel(?alg,?m)

Regarding our running example, we have considered to both cases the same domain objective and working data set:

Modeling objective: classification
workingDataset {
idCard; idCliente; birthDate; age; initialCardDate; cardAge; initialCustomer; clientAge; carClientGap; postCode; postCod3; civilStatus; sex; vehicleType; vehicleYear; vehicleAge; fuel; dFirstTransaction; dLastTransaction; nTransactions; tLiters; tAmountFuel; tShopValue; tQtdShop; 1stUsed; 2stUsed; 3stUsed
}

Firstly we have evaluated the most predictive attributes more suitable for this particular classification problem.

We have used two different statistical approaches: SVM (Support Vector Machine) and Chi-squared test. As result the most feature set (given the similarity between gain ratio and SVM outputs – bold marked) was used in data mining experiments.

hasAttributeEvaluation(SVMA) -> { **idCard**; idCliente; **age**; cardAge; clientAge; **carClientGap; civilStatus; sex; vehicleType; vehicleAge**; fuel; **nTransactions; tLiters; tAmountFuel; tShopValue; tQtdShop; 1stUsed; 2stUsed; 3stUsed** }

hasAttributeEvaluation(gainRatio) -> { **idCard; age; carClientGap**; postCode; **civilStatus; sex; vehicleType; vehicleAge; nTransactions; tLiters; tAmountFuel**; tShopValue**; tQtdShop; 1stUsed; 2stUsed; 3stUsed** }

At the end of this evaluation steps we have considered the following algorithm working data set:

```
workingDataset {
        idCard; age; carClientGap; civilStatus; sex; vehicleType; vehicleAge; nTransactions; tLiters;
                tAmountFuel; tQtdShop; 1stUsed; 2stUsed; 3stUsed
        }
```

Then we have split the algorithm working data set into training set and test set.

```
hasTraningSet= 66,6% (6 183 records)
hasTestingSet= 33,3% (3 102 records)
```

In order to perform the customer classification we have used, among others, four classification algorithms: J48 (C45 implementation[Quinlan1986]); Random Tree; ZeroR and NaiveBayes [Witten and Frank2000].

As target attribute we have settled the same to all of them – *1stUsed* (the first most used oil station), which is a categorical attribute that categorizes the most used oil station according the distance from customer address:

p – Less than 10km;
s - Between 10 and 30km;
t - More than 30 kms.

To build up these mining experiments we have used Weka Toolkit [Witten and Frank2000] which allowed not only the actual mining but also feature analysis and

algorithm evaluation. These experiments did not aim at the full construction of a classification model but instead test and analyze different approaches and further ranking.

Algorithm use is performed through the following code, e.g., J48 parameters settings:

```
hasAlgoritm(alg)→J48
hasAlgorithmParameter (gainRatio) =  workingDataSet(wds) → hasModel(?m)
```

The most relevant rule extracted from above data algorithms use was:

```
if (age<27 and vehicleType=”Lig” and sex=”Female”) then 1stUsed=”p”
```

At this model we may say that, female card owners with less than 27 years old and have a car "*lig*" (ligeiro) category, it would use fuel station located in range than 10 kilometers than her address.

7.5. Deployment

Each running KDD process must be evaluated according to the results, in order to update the knowledge base for a latter reuse. The SWRL code would be in the form:

```
:getEvaluation[(Model?m)^
        hasModeling(?met)^
        hasAlgorithm(?alg)^
        hasEvaluation(?m,?met,?alg) ^
         hasEvaluationParameters(?par)]
                        -> Evaluation (?m,?ev)]
```

Each evaluation depends on e.g., model type or algorithms used.

To answer the question: "*customer profile according fuel station use*" we have got the models which now are being evaluated. In this example we have used the basic Weka algorithm evaluator tables, presented at above Table 1 and Table 2. In order achieve a more evident results we have introduced two more performance index: accuracy

(generally, how good is the model) and sensibility (how good the model is at correct result classification)

Table 1: Algorithm performance evaluation tables

	J48	ZeroR	NaiveBayes	Random Tree
Correctly Classified Instances %	67.41 %	69,07%	68.98%	56.84%
Kappa statistic	0.0018	0	0	0
Mean absolute error	0.427	0.43	0.43	0.43
Root mean squared error	0.48%	0.46	0.47	0.66
Relative absolute error (%)	99.92%	100%	101.16%	101.3%
Root relative squared error (%)	99.92%	100%	101.06%	141.8%

Table 2: Algorithm True Positive (TP)/False positive (FP) evaluation

	J48			ZeroR			Naïve Bayes			RandomTree		
Class	**TP**	**FP**	**PRI**	**TP**	**FP**	**PRI**	**TP**	**FP**	**PRI**	**TP**	**FP**	**PRI**
P	0.955	0.954	0.802	1	1	0.817	0.998	0.998	0.816	0.686	0.693	0.687
S	0.046	0.045	0.008	0	0	0	0002	0.002	0.003	0.307	0.314	0.305

Table 3: Algorithms confusion matrix

	J48	
Class	**P**	**S**
p	6126	286
s	2740	132

	ZeroR	
	p	**s**
	6412	0
	2872	0

	NaiveBayes	
	P	**S**
	6399	13
	2867	5

	RandomTree	
	p	**s**
	4396	2016
	1991	881

Accuracy:	67,41%	69,07	69,8%	56,84%
Sensibility:	95,5%	100%	99,8%	68,6%

Then this (e.g., J48 algorithm) model will be added to the knowledge base as:

INSERT record KNOWLEDGE BASE

hasAlgorithm(J48) AND
hasModelingObjectiveType(classification) AND
hasAlgorithmWorkingData ({idCard; age; carClientGap; civilStatus; sex;
vehicleType; vehicleAge; nTransactions; tLiters; tAmountFuel;
tQtdShop; 1stUsed; 2stUsed; 3stUsed }) AND
Evaluation(67,41%; 95,5%) AND
hasResultMoldel (J48;classification; “wds”,PCC;0,84;0,29)

Once performed the evaluation, the system automatically updates the knowledge base with a new record. The registered information will serve for future use – knowledge sharing and reuse. Moreover, ontology is also being evaluated through the index precision and recall.

8. Discussion

This section provides a further discussion on the application of our approach to assist the KDD process. Our system prototype operation follows general KDD framework as presented in Figure 1. At this point we aim to answer: How much could the KDD process be improved through ontology assistance?

Since the ontology contribution to the KDD process is quantitatively uncertain we have used a quality approach based on KDD team individual expert contribution (Table 1). Thus, we have classified into two states (plus and minus relevant) each team individual. Positive symbol (⊕) refers to someone which is in charged to perform or to participate in KDD phase; with negative (⊖) we refer someone which may be participate but it is so much relevant his/her presence. At the first sight some comments to appear relevant: domain expert has active participation during the former KDD phases (objective definition and data comprehension) and at the effective model evaluation – deployment. Database expert has relevant role during the central KDD phases, from data understand to modeling phases. Finally, data analyst expert, as the one in charge to perform operational tasks throughout algorithms use in order to find relevant information.

Table 4: KDD team elements participation without ontology assistance

Without Ontology	Objective	Data Understanding	Data Pre-Processing	Modeling	Evaluation	Deployment
Domain Expert	⊕	⊕	⊖	⊖	⊖	⊕
Database Expert	⊖	⊕	⊕	⊕	⊖	⊖
Data Mining Expert	⊖	⊖	⊕	⊕	⊕	⊕

Our quality assessment to ontology assistance to KDD process is based on comparison between the work load in a process with and without ontology. Using a qualitative open

questions questionnaire, we have analyzed the work load to each KDD team element and judge accordingly if his work has been positively reduced (⊕), remains equal (=) or by contrary it has augmented (⊖).

Table 5: KDD process with ontology assistance

With Ontology	Objective	Data Understanding	Data Pre-Processing	Modeling	Evaluation	Deployment
Domain expert	⊕ Effective scope definition	⊕ Data hands-on systematization	=	=	=	=
Database expert	=	=	⊕ Attribute handling insights	⊕ Useful to algorithm data preparation	=	=
Data analyst expert	=	=	⊕ Systematize tasks	⊕ Algorithm selection Data preparation insights	⊕ Evaluation methods improvement	=

Noticeably we did not find any negative quote to the ontology assistance. Moreover, observing the Table 5 we note to all participants a common understand about ontology assistance whenever they participation to the process is ranked as more relevant (in Table 4). That is, as much is the user involvement in KDD process, as much does the ontology assistance is more useful. Here, the positive symbol (⊕) rate in Objective and Data Understanding phases.

Nevertheless the successful ontological assistance role at KDD, it is very human dependent process. Indeed, ontologies provide much domain or tasks information at each KDD phase. However this information use will always be human dependent, as deployment phase proves, which, accordingly our experts did not registered any ontological improvement.

9. Conclusions and further research

The KDD success is still very much user dependent. Though our system may suggest a valid set of tasks which better fits in KDD process design, it still miss the capability of automatically runs the data, develop modeling approaches and apply algorithms.

This work strived to improve KDD process supported by ontologies. To this end, we have used general domain ontology to assist the knowledge extraction from databases with KDD process.

This research focuses the KDD development assisted by ontologies. Moreover we use ontologies to simplify and structure the development of knowledge discovery applications offering to a domain expert a reference model for the different kind of DM tasks, methodologies to solve a given problem, and helping to find the appropriate solution.

There are four main operations of KDD that can take advantage of domain knowledge embedded in ontologies:

i. During the data preparation phase, ontology can facilitate the integration of heterogeneous data and guide the selection of relevant data to be mined;
ii. During the mining step, domain knowledge allows the specification of constraints for guiding DM algorithms by, e.g. narrowing the search space;
iii. During the deployment phase, domain knowledge helps experts to validate extracted units and ranking them.
iv. With knowledge base ontology may help analyst to choose the best modeling approach based on knowledge base ranking index.

Future work will be devoted to expand the use of KDD ontology through knowledge base population with more relevant concepts about the process. Another interesting direction to investigate is to represent the whole knowledge base in order to allow its automatic reuse.

References

[Agrawal *et al.*1993] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In on Management of Data, I. C., editor, SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international

conference on Management of data, pages 207–216. International Conference on Management of Data.

[Anand *et al.*2007] Anand, S. S., Grobelnik, M., Herrmann, F., Hornick, M., Lingenfelder, C., Rooney, N., and Wettschereck, D. (2007). Knowledge discovery standards. Artificial Intelligence Review, 27(1):21–56.

[Bernstein *et al.*2005] Bernstein, A., Provost, F., and Hill, S. (2005). Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. IEEE Transactions on knowledge and data engineering, 17(4).

[Blazquez *et al.*1998] Blazquez, M., Fernandez, M., Garcia-Pinar, J. M., and Gomez-Perez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. In Knowledge Acquisition Workshops and Archives, Voyager Inn, Banff, Alberta, Canada. University of Calgary.

[Brezany *et al.*2008] Brezany, P., Janciak, I., and Tjoa, A. M. (2008). Data Mining with Ontologies: Implementations, Findings, and Frameworks, chapter Ontology-Based Construction of Grid Data Mining Workflows, pages 182–210. Information Science Reference - IGI Global.

[Chu and Hwang2008] Chu, H.-C. and Hwang, G.-J. (2008). A delphi-based approach to developing expert systems with the cooperation of multiple experts. Expert Systems with Applications, 34:2826–2840.

[Delbecq *et al.*1975] Delbecq, A. L., Ven, A. H. V. D., and Gustafson, D. H. (1975). Group Techniques for Planning- A Guide to Nominal Group and Delphi Processes. Scott.

[Domingos2003] Domingos, P. (2003). Prospects and challenges for multi-relational data mining. SIGKDD Explorer Newsletter, 5(1):80–83.

[Fayyad *et al.*1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. In Magazine, A., editor, AI Magazine, volume 17, pages 37–54, Univ Calif Irvine, Dept Comp & Informat Sci, Irvine, Ca, 92717 Gte Labs Inc, Knowledge Discovery Databases Kdd Project, Tech Staff, Waltham, Ma, 02254. American Association for Artificial Intelligence.

[Fayyad and Uthurusamy1996] Fayyad, U. and Uthurusamy, R. (1996). Data mining and knowledge discovery in databases. In ACM, editor, Communications of the ACM, volume 39, pages 24–26, New York, NY, USA. ACM.

[Fernandez *et al.*1997] Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. Technical report, AAAI.

[Friedman-Hill and Scuse2008] Friedman-Hill, E. and Scuse, D. (2008). Jess: The rule engine for the java platform. Technical report, Sandia National Laboratories.

[Gomez-Perez *et al.*2004] Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). Ontological engineering. Springer, 2nd edition.

[Gottgtroy *et al.*2004]Gottgtroy, P., Kasabov, N., and MacDonell, S. (2004). An ontology driven approach for knowledge discovery in biomedicine.

[Gottgtroy *et al.*2003]Gottgtroy, P., Kasabov, N., and Macdonelll, S. (2003). An ontology engineering approach for knowledge discovery from data in evolving domains. Technical report, Knowledge Engineering and Discovery Institute, Auckaland University of Technology - New Zealand.

[Gruber1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199–220.

[Han and Kamber2001] Han, J. and Kamber, M. (2001). Data mining: concepts and techniques. Morgan Kaufman, San Francisco, CA.

Holger Knublauch, Ray Fergerson, Natalya Noy, and Mark Musen. The protege owl plugin: An open development environment for semantic web applications. In The Semantic Web ISWC 2004, pages 229–243. Springer, 2004.

[Horrocks *et al.*2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). Swrl: A semantic web rule language - combining owl and ruleml. Technical report, W3C.

[McBride2002] McBride, B. (2002). Jena: A semantic web toolkit. IEEE Internet Computing, 6(6):55–59.

[Nigro *et al.*2008] Nigro, H. O., Cisaro, S. G., and Xodo, D. (2008). Data Mining with Ontologies: Implementations, Findings and Frameworks. Information Science Reference. Information Science Reference - IGI Global, London, igi global edition.

[Parsia and Sirin2004] Parsia, B. and Sirin, E. (2004). Pellet: An owl dl reasoner. In 3rd International Semantic Web Conference (ISWC2004).

[Phillips and Buchanan2001] Phillips, J. and Buchanan, B. G. (2001). Ontology-guided knowledge discovery in databases. In ACM, editor, International Conference On Knowledge Capture 1st international conference on Knowledge capture, pages 123–130. International Conference On Knowledge Capture.

[Pinto *et al.*2009a] Pinto, F. M., Gago, P., and Santos, M. F. (2009a). Marketing database knowledge extraction â€“ towards a domain ontology. In IEEE 13th International Conference on Intelligent Engineering Systems 2009.

[Pinto *et al.*2009b] Pinto, F. M., Marques, A., and Santos, M. F. (2009b). Database marketing process supported by ontologies: System architecture proposal. In 11th International Conference on Enterprise Information Systems.

[Quinlan1986]Quinlan, R. (1986). Induction of decision trees. Machine Learning, 1:81–106.

[Sarwar *et al.*2000] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In Proceedings of the second ACM conference on electronic commerce.

[Seaborne2004] Seaborne, A. (2004). RDQL - A query language for rdf. Technical report, W3C.

[Smith and Farquhar2008] Smith, R. G. and Farquhar, A. (2008). The road ahead for knowledge management: An ai perspective. American Association for Artificial Intelligence, 1:17–40.

[Witten and Frank2000] Witten, I. H. and Frank, E. (2000). Data Mining: Practical Machine Learning Tools and Technique. The Morgan Kaufmann Series in Data Management Systems, 2nd edition.

4.4 Database Marketing Intelligence Supported in Ontologies and Knowledge Discovery in Databases

4.4.1 Introduction

This PhD research aims to present a prototype of Database Marketing Intelligence (DBMI) Methodology Based on Ontologies and Knowledge Discovery in Databases. Accordingly, we have developed worked on ontologies and knowledge discovery in databases attaining to get an ontological knowledge base, a systemic approach within a general framework perspective - the DBMI methodology proposal. This will be used for leveraging the semantic content of ontologies to guide the entire DBM process. Indeed, we have designed a methodology (DBMI) which based on KDD role and guided by ontologies (presented in previous sections) fulfils the entire DBM process.

The main goal of the DBMI is to assist the user in carrying out the KDD process within DBM projects. The DBMI provides support in choosing particular knowledge extraction objectives and manage the entire process, from data evaluation to output models evaluation. Also, DBMI dynamically modifies the tasks' composition depending on the marketing activity objectives, entered data, defined preconditions and effects, and existing description of services available in the knowledge base.

4.4.2 Research approach

We have adopted an action research methodology approach to the system prototype design and development. Action research is a self-reflective, self critical and critical enquiry undertaken by professionals to improve the rationality and justice of their own practices, their understanding of these practices and the wider contexts of practice (Lomax, 2002). Moreover, action research methodology contributes to the development and improvement of systems. This methodology incorporates the four-step process (Figure 11) of planning, acting, observing and reflecting on results from a particular

project or body of work (Zubber-Skerritt, 2000), (O'Brien, 2002). The concept is essentially concerned with a group of people who work together to improve their work processes (Baskerville, 1999), (Carson *et al.*, 2004).

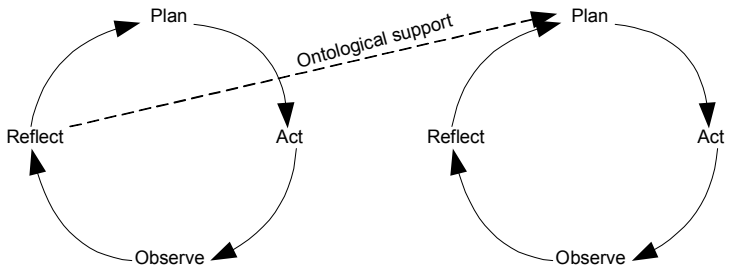


Figure 14: Action research methodology

This choice of action research was based on two reasons. First, due to the low number of scientific research works that have been conducted on supporting KDD process over intelligent structures like ontologies. Second, ontologies can play an important role in the knowledge development as long as they register past knowledge for future reuse (Figure 11). Thus exploratory research was required and action research provides this capability better than many other alternatives (Dick, 2008).

Nevertheless, first we need to formulate, test, deploy and evaluate a complete DBM project whereas DBM is developed over a KDD process. Hence, we focus such interaction and annotate it in a semantic language, like RDF (Resource Description Framework) or OWL (Ontology Web Language) and use the SWRL (Semantic Web Language Rule) to infer.

4.4.3 Results

Since our work focuses on the integration of knowledge discovery techniques and ontologies at database marketing process, we have developed a DBMI prototype whereas DBM process is developed using KDD process for knowledge extraction and ontologies for improvement and assistance. Indeed, the main goal of the DBMI is to assist the user in carrying out the KDD process in the DBM process.

DBMI provides support in choosing particular knowledge extraction objectives and managing the entire process, from data to output models evaluation. Moreover, DBMI dynamically modifies the tasks composition depending on marketing activity objectives, entered data, defined preconditions and effects, and existing description of services available in the knowledge base, as presented in next section.

4.4.4 Database Marketing Intelligence Supported by Ontologies

Published in:

WSEAS Transactions on Business and Economics
World Scientific and Engineering Academy and Society
volume 6, pages 135-146
March 2009.

Abstract— In this work we use ontologies at an almost unexplored research area within the marketing discipline, throughout ontological approach to the database marketing. We propose a generic framework supported by ontologies for the knowledge extraction from marketing databases. Therefore this work has two purposes: to integrate ontological approach in Database Marketing and to propose domain ontology with a knowledge base that will enhance the entire process at both levels: marketing and knowledge extraction techniques.

1. Introduction

Technology has provided marketers with huge amounts of data and artificial intelligence researchers with high level processing rate machines. At the marketing practice we note that marketing database is normally used in organizational secret and closed purpose, which limits the knowledge for reuse and sharing. Database Marketing (DBM) is a database oriented process that explores database information in order to support marketing activities and/or decisions. The Knowledge Discovery in Databases (KDD) process is well established amongst scientific community as a three phase process: data preparation, data mining and deployment/evaluation. The KDD has been successfully applied in various domains particularly in the marketing field. Nevertheless, previous well established concepts and scientific dominance regarding each one of these methods seem to have a lack of knowledge concerning its application amongst different requirements and conditions.

2. Ontologies at Marketing Field

Ontologies are becoming, nowadays, one of the most popular knowledge representation techniques. When ontologies are formalized in any kind of logic representation, they can also support inference mechanisms [Jurisica *et al.*1999]. For a given collection of facts, these mechanisms can be used to derive new facts or check for consistency. Such computational aids are clearly useful for knowledge management, especially when dealing with complex and heterogeneous knowledge problems or with large amounts of knowledge.

Ontologies, model the structure of data (e.g., representing sets of classes and their properties or attributes), the semantics of data (e.g., in the form of axioms that express constraints such as inheritance relationships, or constraints on properties), and data instances (often called individuals). Ontologies use a formal domain or knowledge representation, agreed by consensus and shared by an entire community [Jurisica *et al.*1999]. To integrate ontologies, we must be able to understand the relationship between structures and data in different ontologies. (Figure 1)

Ontologies roles in DBM may have particular significance due its cross research (both marketing and extraction techniques knowledge is needed) area focus. Indeed, ontologies can play an important role describing in a semantic form, all concepts and techniques around the process[Zhou2007]. Moreover, with such description it will also be possible, to introduce metrics to compare and therefore select and suggest the best approaches and methods to a new project.

An ontology defines the vocabulary used to compose complex expressions such as those used to describe resource constraints in a planning problem [Gomez-Perez *et al.*2004]. Here is the main reason why vocabulary is one of the focus of ontological commitments.

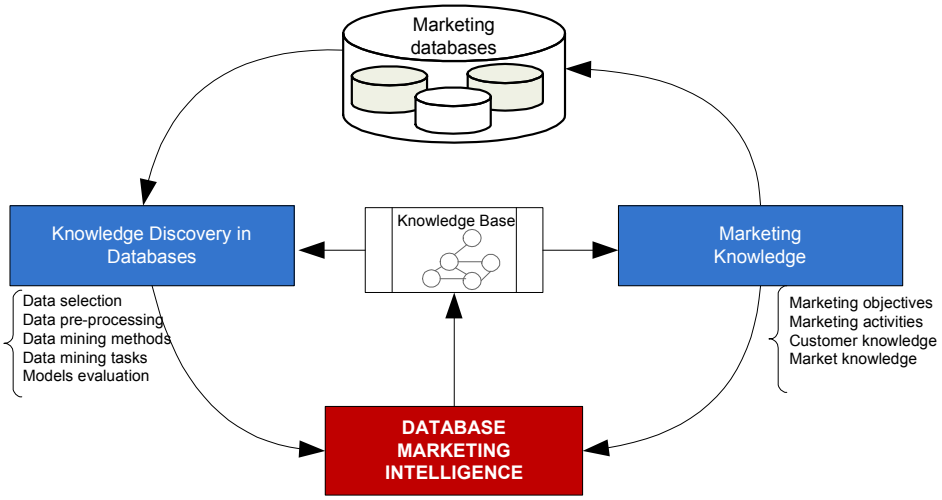


Figure 1: Database marketing intelligence general modules perspective

Our DBMI system proposal has two main input blocks (KDD and marketing knowledge). KDD feeds the ability (e.g., methods, techniques or tasks) to extract useful information from marketing databases; Marketing knowledge provides all input variables (e.g., objectives, activities, or other domain knowledge) necessary to the DBM process development. Therefore, DBMI uses ontologies and update them.

3. Applied Engineering Methodology

The Database Marketing Ontology (DBMO) has been developed according two methodological principles (adapted from [Jarrar2008]):

i) *ontology domain double articulation*: this principle suggests that an ontology is doubly articulated into: domain axiomatization and application. To capture knowledge at the domain level, one should focus on characterizing the intended meaning of domain concepts. Through this articulation method, our work developed a cross-research between marketing concepts (objectives and activities) and knowledge extraction techniques. As example, thereafter marketing knowledge (marketing objectives and marketing activities) double articulation proceeds through axiomatization e.g., sentences that states the necessary and sufficient conditions for e.g., some algorithm condition

development in terms algorithm type definition, modeling objectives and data type required.

ii) *Ontology modularization*: The modularization principle suggests that applications should be built in a modular approach. It combines all axioms introduced in the composed modules (here each module refers to database marketing process phase). Thus, modules will make the ontology maintenance and reuse easier.

Therefore, according to the double principle, the ontology has a knowledge axiom structure that reflects a structured marketing knowledge and also a structured database marketing knowledge. Hence, our approach has been developed at two strands: marketing knowledge and database oriented knowledge extraction process.

4. Knowledge Base

A synergy between decision support systems and knowledge extraction process management is possible [Bolloju *et al.*2002]. Ontologies can play an important role in this area, throughout domain knowledge and process management integration. Here we introduce the knowledge base role (Figure 2). This ontological layer is the main core of the system, whenever a new DBM process starts; it both suggests from previous related marketing knowledge and registers according taken options.

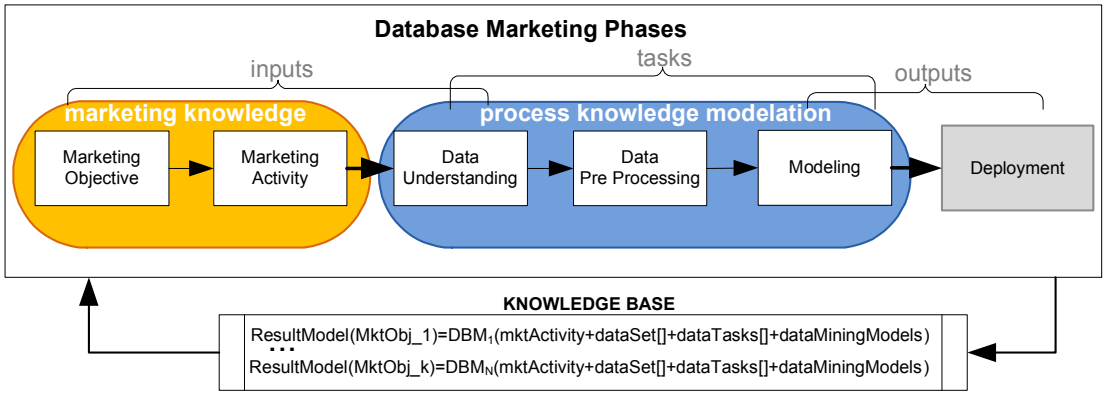


Figure 2: Supported database marketing intelligence process phases

The efficiency of the interaction between marketing objectives, marketing databases and knowledge extraction process is mainly based on the instantiation process at the knowledge base. This process is fully data dependent on integration issues and has to be controlled by the domain expert, who has to choose the most accurate and valid information related to each case.

In short, knowledge base holds all DBM related process, during the instantiation process by adding a new record. Each record in knowledge base refers each running database marketing process.

KNOWLEDGE BASE RECORD
{ Marketing objectives
Marketing activity
Data used (selection)
Data preparation tasks
Algorithms used
Model description
Analytical model evaluation
Model optimization
Business decision
}

5. System Architecture Proposed Solution

Our solution integrates formal (marketing field) and database extraction process (extraction process) knowledge. Indeed, our architecture proposal defines relations and constrains between input elements (e.g., data items, data or modeling tasks) and DBM outputs (models and related evaluation) through a knowledge base instantiation (Figure 3).

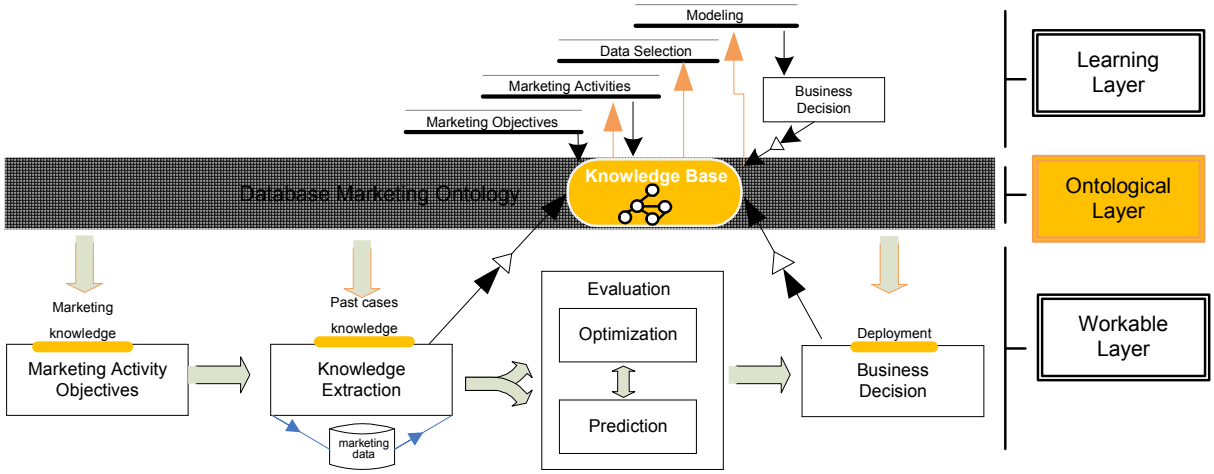


Figure 3: system architecture proposal

The DBMI architecture is a layered structure as shown in Figure 3. The object-oriented design gives the system flexibility and expandability. It consists of three layers:

- Learning layer: this metadata layer has in charge both, to assist at the DBM process and update the knowledge base whenever an action is taken;
- Ontological layer: A synergy between decision support systems, knowledge management is possible [Bolloju *et al.*, 2002]. Ontologies can play an important role in this area, integrating both previous and proceeding to the decision moment. This layer is the main core of the system positioned at the middle between physical and operational layers. Whenever a new DBM process starts, it both suggests and register (according Table 1):

▪ *Registering task* is developed according a relational database structure schema previously defined. Relevant information is registered within those tables with specific rules. Those tables form the knowledge base, which as the ability to organize and systematize DBM process information. Moreover, has the capability to use, compute and provide information in an actionable way to the user needs.

▪ *Suggestion task* is performed using previous information in the knowledge base. Ontology has the capability to query the knowledge table with previous user loaded information.

Table 2 ontological interface operational framework

i)	Register a new DBM instance indentifying marketing objectives;
ii)	Suggests which marketing activities can be possible to be developed
iii)	Register selected marketing activity to be developed;
iv)	Suggests which data should be loaded to attain the marketing activity objective, classifying the data it in two major classes: indispensable and useful
v)	Record all data available and classify hose attributes in terms of main marketing data type categories;
vi)	Suggest data tasks to be developed
vii)	Register data tasks developed by the user according the previous classification;
viii)	Suggest which models best fits to achieve activity objectives;
ix)	Register algorithm selected;
x)	Suggest data pre-processing and data preparation specific tasks to be performed;
xi)	Register algorithm results, in terms of model description and technical evaluation
xii)	Register business decision and evaluation

The interaction between workable (physical) and ontological occurs whenever a step forward in the DBM process is done. To the first register task is performed and to the last suggesting action is deployed.

- Workable layer or user interaction bridge layer. Here we consider the ontological assistance work and all related operations carried by user during the DBM process. We consider the DBM process into four main phases: Marketing activity objectives, knowledge extraction, evaluation and business decision, as follows:

  - *Marketing activity definition*: The role of ontologies in business understanding is not peculiar to marketing discipline. Domain ontologies are an important vehicle to inspect a domain prior to committing to a particular task. Semi-formal ontologies can help a newcomer to get familiar with most important concepts and relationships, while formal ontologies allow the identification of conflicting assumptions that might not be obvious at the first sight;
  - *Knowledge extraction*: For improved data exploration, elements of ontology have to be (presumably manually) mapped onto elements of the data scheme and vice versa. This will typically lead to selecting a relevant part of ontology (or multiple ontologies) only. Another relevant issue is the connection between the Data Preparation phase and the subsequent Modelling phase. Concrete use of domain ontology depends partially on the chosen mining tool/s. An ontology may typically help by identifying multiple groups for attributes

and/or values according to semantic criteria. In the Modelling phase, ontologies might help design the individual mining sessions. In particular for large datasets, it might be worthwhile to introduce some ontological bias, e.g., to skip the quantitative examination of hypotheses that would not make sense from the ontological point of view, or, on the other hand, of two obvious ones;

- *Evaluation phase*: the discovered model/s have the character of structured knowledge built around the concepts (previously mapped on data attributes) and can be interpreted in terms of the ontology and associated background knowledge;
- In the *Business Decision phase*: extracted knowledge is fed back to the business environment. Provided we previously modeled the business using ontological means, the integration of new knowledge can again be mediated by the business ontology. Furthermore, if the mining results are to be distributed across multiple organizations (say, using the semantic web infrastructure), mapping to a shared ontology is inevitable.

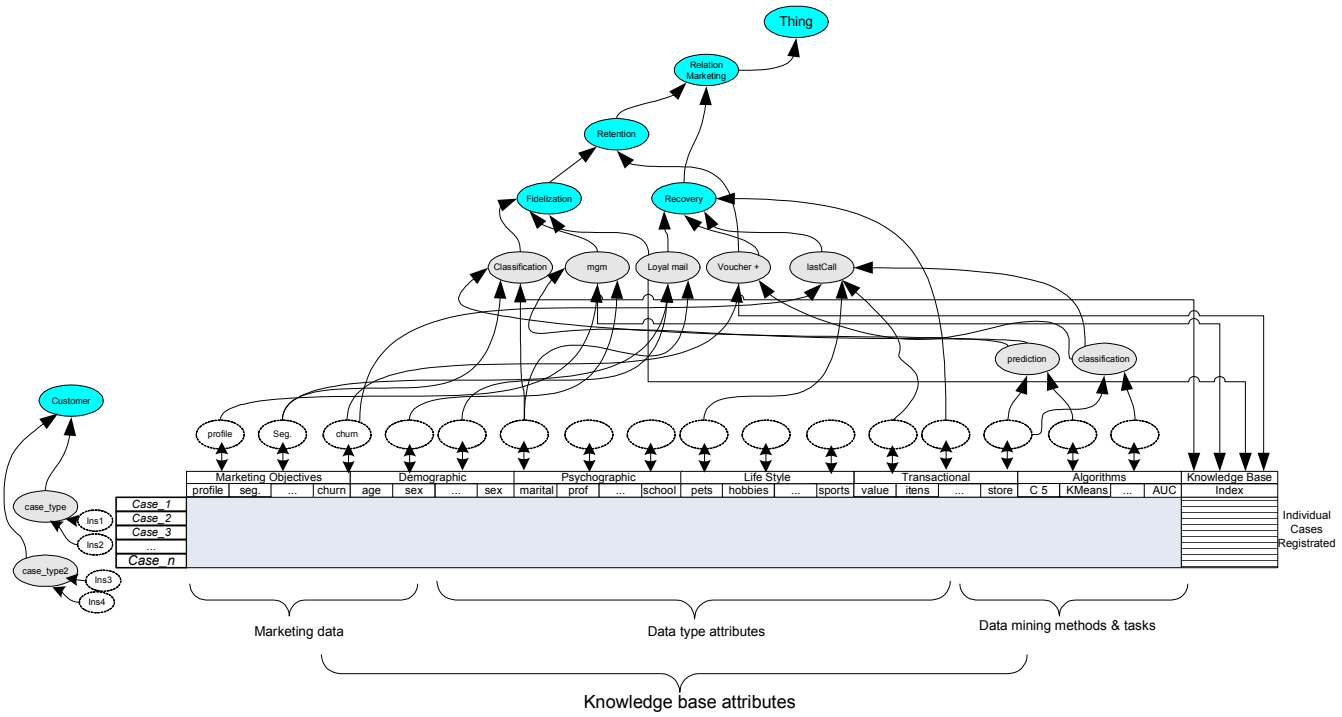


Figure 4: Articulation between data and knowledge at DBM ontology

Our system operation is based on a "mapping" between objects and attributes of the dataset, and instances of the knowledge base (Figue 4). Thus, formalized knowledge within the knowledge base can be used for guiding DBM process across all phases.

Figure 4 illustrates this attribute mapping in the case of database marketing variables such as: data type information, marketing objectives or KDD methods and tasks. The efficiency of the interaction between data and knowledge is mainly based on the instantiation process in the knowledge base with collected data. This process is dependent on data integration issues and has to be controlled by the domain expert, who has to choose the most accurate class corresponding to the considered data. In this way, the domain expert is in charge of instantiating the right classes in the knowledge base. Moreover, our knowledge base proposal has the ability to register all past DBM developed projects.

6. Conclusions

This ontological DBM approach solution appears promising for both marketers and computer scientists. One of the promising interests of DBM ontologies is its use for guiding the process of knowledge extraction from marketing databases. Indeed, one of the promising interests of DBM ontology is its use for guiding the knowledge extraction process from marketing databases.

This idea seems to be much more realistic now that semantic web advances have given rise to common standards and technologies for expressing and sharing ontologies [Coulet *et al.*, 2008] [Smith *et al.*, 2008]. Therefore our DBMI can take advantage of domain knowledge embedded in ontologies as follows:

i) at the marketing activity definition, ontology can indicate a global perspective which is possible to do or not to do with the available resources, e.g., based on data completeness or heterogeneity;

ii) from a DBM objectives point of view, ontology may suggest or select the most appropriate approaches to treat the available data;

iii) during the data preparation step, DbmO can facilitate the integration of heterogeneous data and guide the selection of relevant data to be used;

iv) at the modelling phase (e.g. data mining), domain knowledge allows specification of constraints to guide data mining algorithms by, e.g., narrowing search space;

v) during the interpretation step, domain knowledge helps experts to visualize and validate extracted units

Therefore, marketing domain is an interesting and challenging domain for representation and ontology development. Further research on this subject will be aimed in both ways, with the final task of comparing diversities between OWL and Frames ontology and showing their advantages and disadvantages for a subject domain.

References

[Bolloju *et al.*2002] Bolloju, N., Khalifa, M., and Turban, E. (2002). Integrating knowledge management into enterprise environments for the next generation decision support. Journal of Decision Support Systems, 33(2):163–176.

[Coulet *et al.*, 2008] Coulet, A., Smail-Tabbone, M., Benlian, P., Napoli, A., & Devignes, M.-D. (2008). Ontology-guided data preparation for discovering genotype-phenotype relationships. Journal of BMC Bioinformatics, 9, 1–9.

[Gomez-Perez *et al.*2004] Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). Ontological engineering. Springer, 2nd edition.

[Jarrar2008] Jarrar, M. (2008). Towards Effectiveness and Transparency in e-Business Transactions, An Ontology for Customer Complaint Management, chapter An Ontology for Customer Complaint Management. Idea Group Inc.

[Jurisica *et al.*1999] Jurisica, I., Mylopoulos, J., and Yu, E. (1999). Ontologies for knowledge management: An information systems perspective. In for Information

Sciences, A. S., editor, Proceedings of the Annual Conference of the American Society for Information Sciences (ASISâ€™99). American Society for Information Sciences.

[Smith *et al.*, 2008] Smith, M. K., Welty, C., & McGuinness, D. L. (2008). OWL Web Ontology Language Guide. W3C.

[Zhou2007] Zhou, L. (2007). Ontology learning: state of the art and open issues. Information Technology and Management, 8:241–252.

4.5 Publications

Throughout this PhD program, in order to present and get feedback from scientific community some work has been published:

Filipe Mota Pinto, Alzira Marques, and Manuel Filipe Santos. *Ontology-supported database marketing*. Journal of Database Marketing & Customer Strategy Management, 16:76–91, 2009.

Filipe Mota Pinto, Pedro Gago and Manuel Filipe Santos. *Marketing database knowledge extraction: towards a domain ontology*. In IEEE 13th International Conference on Intelligent Engineering Systems 2009, 2009.

Filipe Pinto, Manuel Filipe Santos, and Alzira Marques. *Ontology based Database Marketing – A contribution to Business Intelligence*, Proceedings from 10th International Conference on Mathematics and Computers in Business and Economics (MCBE'09) Prague 2009.

Filipe Mota Pinto, Alzira Marques, and Manuel Filipe Santos. *Database marketing process supported by ontologies*: System architecture proposal. In 11th IEEE International Conference on Enterprise Information Systems, 2009.

Filipe Pinto, Alzira Marques, and Manuel Filipe Santos. *Ontological approach to database marketing workflow framework*. In Global Business and Technology Association, Eleventh Annual International Conference, Prague, Czech Republic, 2009.

Filipe Pinto, Manuel Filipe Santos, and Alzira Marques. WSEAS Transactions on Business and Economics, volume 6, chapter *Database marketing intelligence supported by ontologies*, pages 135–146. World Scientific and Engineering Academy and Society, 2009.

Teresa Guarda and **Filipe Pinto**. *Data pre-processing issues: a case study for database marketing*. In Conferencia Iberica em Sistemas de Informação (CISTI), 2009.

Filipe Pinto, Manuel Filipe Santos and Alzira Marques. *Ontological Assistance for Knowledge Discovery in Databases*, WSEAS Transactions on Information Science and Applications, World Scientific and Engineering Academy and Society, (accepted for publication in press).

Filipe Pinto, Alzira Marques and Manuel Filipe Santos. *Data mining approach in relationship marketing database*. In Global Business and Technology Association, Tenth Annual International Conference, July 2008.

Filipe Pinto, Alzira Ascenção Marques and Manuel Filipe Santos. *Customer insights from transactional database: Database marketing case*. In International Conference on Information Systems - Data Mining 2008, 2008.

Filipe Mota Pinto, Alzira Marques and Manuel Filipe Santos. *Customer knowledge from transactional database: making the data work*. Journal of Sinoeuropean Engineering Research Forum, 1:34–39, 2008. ISSN 1757-4307.

Filipe Pinto, Pedro Gago and Manuel Filipe Santos. *Data mining as new paradigm for business intelligence in database marketing projects*. In Enterprise Information Systems, ICEIS 2006, Proceedings of Artificial Intelligence and Decision Support Systems, 2006.

Filipe Pinto, Pedro Gago and Manuel Filipe Santos. *Database marketing as a support to marketing activities*. In Global Business and Technology Association, Tenth Annual International Conference, 2006.

Articles in revision process

Journal of Data Knowledge and Engineering - *Ontological Approach to Marketing Databases Exploration: Towards to Database Marketing Ontology*

Journal of Integrated Marketing Communication - *Ontologies Role at Database Marketing*

Articles under reviewing

International Journal on Semantic Web & Information Systems - *Ontological Engineering for Knowledge Discovery in Databases Process*

V

Discussion and Conclusions

In this chapter we present a synopsis regarding all research work, a discussion focusing the developed work, its drawbacks, limitations and advantages, and, at the end, a synthesis of all contributions and some further work considerations.

5.1 Synopsis

Subsequently to some previous research work, we have proposed as main objective for this PhD work, to study and evaluate, *"how ontologies may facilitate the process of knowledge discovery from databases with special focus within database marketing field"*. Such objective comes from the evidence that too much work has been done in KDD area but also very little knowledge sharing towards an automatic (or at least semi-automatic) KDD assisted process has become available. Since ontologies attain to share and reuse knowledge, we consider them as a promising for this area.

Therefore we have designed our research in the DBM supported by ontologies and KDD process. Accordingly, the study of different approaches for DBM throughout knowledge extraction methods and techniques is aimed at enhancing the competence of ontologies as an assistance guide. For this end, the following tasks have been carried out:

Knowledge Discovery in Databases Ontology (presented in section 4.1) – we have developed an ontology that focuses the general KDD process, in order to formalize all related knowledge regarding all phases, methods and tasks. Thus, this ontology will aid as a general guide to data analyst along the running KDD process;

Database marketing process ontology (presented in section 4.2) – this ontology holds two different knowledge structure knowledge concepts, relations and properties, from two different scientific areas: database marketing and KDD. In order to propose this ontology, we have reused the former KDD ontology, since our referential data analyses module within DBM framework is supported by KDD process;

Ontological Assistance to KDD (presented in section 4.3) – Once KDD ontology has been created, we have performed a practical knowledge discovery running process over a real marketing database. Moreover, with this work, we have demonstrated how ontologies can effectively assist the KDD process in each phase;

Database marketing intelligence methodology framework (presented in section 4.4) – our main general objective focuses the use of ontologies in order to assist the KDD process at the DBM field. Therefore, we have developed a methodology for database marketing supported by ontologies and the knowledge discovery process.

The following research targets have been achieved in particular:

(i) Marketing knowledge concept structure - through elicitation method we have modeled information about marketing databases exploration processes, that have conducted us to the Database Marketing Ontology;
(ii) Taxonomy construction of knowledge extraction process from databases;
(iii) Development of a database marketing supported by KDD framework - Database Marketing Intelligence;
(iv) Development of a system prototype to the effective KDD ontological assistance.

Since this work bridges three disciplines: ontologies, technologies and information systems; and database marketing, in each work step we have produced scientific reports in the form of articles, conference proceedings, technical reports or progress reports (as presented in the previous section).

5.2 Discussion

This scientific research was developed towards to the effective ontological support to the knowledge discovery in databases in the context of database marketing.

Throughout this document, we have presented the performed research focusing general ontologies creation and database marketing intelligence methodology proposal. Here, the mission of a general ontology is to represent the real world and to facilitate the knowledge reuse and exchange.

With the knowledge discovery ontology we attained to generally describe all related process knowledge and then to formalize methods and tasks. Using the literature review method for knowledge elicitation, we have achieved the necessary background for the complete KDD process formalization. Then we have followed the methontology framework for the ontology construction. This proposal is defined as a domain ontology, which intends to effective support and improve the KDD process in each phase, by suggesting methods and tasks. Such support will enhance the KDD process development in terms of development speed and task selection accuracy. Since KDD process handles so many scientific specifications, such as, data understanding approaches, data preparation objectives or algorithm terms and parameters and, we did not plan to create a full KDD ontology. Therefore, this ontology scope is limited to the knowledge extraction phases main tasks and methods definition.

The database marketing ontology was started from scratch. Indeed, due to the lack of related work in this area, we had to collect the domain concepts from academic and professional experts through the Delphi method. Since we have found the collected concepts valid and consensual among the expert panel, we have started with ontology development following the 101 methodology. This methodology for ontology development is well suited for ontologies reuse. The developed ontology has the scope to aid practitioners throughout the entire DBM process. However, since this ontology reuses the former KDD ontology it remains as general purpose.

Nevertheless, limited ontologies will always be useful for applications in highly specialized domains (Sowa, 2000). Besides, we have also demonstrated how the ontology is aligned with practical cases, throughout an exhaustive step-by-step explanation of both DBM and KDD processes.

Using previously developed ontologies we have proposed a general methodology for database marketing intelligence supported by ontologies and knowledge discovery in databases. The major feature in our approach is that we have used ontologies to effectively assist the KDD process in the DBM process. The major drawback of our approach is that, we have used a general ontologies approach for the database marketing process and therefore, our approach is limited to the general DBM phase (and KDD) assistance work.

The methodology evaluation was performed in two main steps: the ontology structural evaluation and the systematic validation throughout a practical case study. During the former step of the evaluation process, the ontology was submitted to specialists (members of Delphi expert panel) for concept knowledge tree evaluation. In order to prove our proposal, a systematic validation was performed using a real marketing database. We have performed some complete KDD interactions aiming to get some marketing objectives, such as card owner profile, oil station card use profile, or vehicle type card use profile, among others. In fact, even if the application studied is specific, the architecture presented may serve as a basis for any DBM development.

Since our methodology proposal was conceived towards a specific KDD process framework and with a consensual but limited marketing knowledge, one question may arise: how can we use the proposed methodology for the complete and effective assistance to the KDD process at any DBM project? There should not be such limitation. However, such a challenge requires another research dimension and scope project scale.

5.3 Conclusions

During this dissertation we have introduced process oriented ontology for database marketing knowledge based on KDD process. Instead of imposing a fixed order for the DBM process, we have proposed a methodology based on the ontologies and the knowledge extraction process. This methodology is useful since it is used for end user assistance in the entire process development.

The proposed DBMI methodology defines, at different levels, a connection between ontology engineering and KDD process. It also defines a hybrid life cycle for the DBM process, based on both approaches. This life cycle that effectively assists the end-user, is composed by the knowledge extraction process phases and other specific marketing domain activities. Each phase is divided in tasks, directly or indirectly, related to ontology engineering, marketing and KDD.

In spite of important limitations, such as multidisciplinary research scope, the short time frame for its development, this dissertation as produced effective contributions into the three research areas:

- In the knowledge extraction process, a general KDD process ontology which has the ability to assist and suggest at each KDD phase by recommending tasks and methods to be used, was proposed;
- For the ontologies field, we have successfully made ontology integration and reuse, through the integration of general KDD process ontology into the DBM ontology. Indeed our DBMO reuses the former general KDD ontology;
- Also regarding ontologies, we have carried out an original knowledge elicitation method for concept tree construction in a domain where concepts are still not systematized and their definition was not consensual. Indeed, we have adopted the Delphi method within the ontology process development. This ontology development was started from scratch. The Delphi method, was used to achieve consensual knowledge from a community whereas such a subject is almost unknown;

- For the marketing field we have successfully made an innovative ontological engineering approach regarding the database marketing process. We have introduced an ontology assisted framework. To achieve this, a general process framework based on ontologies assistance and KDD was proposed and deployed.

Lastly, this research has focused on finding innovative ways of ontological assistance in practical processes. For this, we have developed and introduced two different ontologies within a general framework. Besides the specific domain knowledge of each, both ontologies provide a vocabulary with a set of concepts and properties for process modeling at operational level. Hence, the main outcome of this research work was the demonstration of how ontologies can be used to assist the KDD process through a knowledge base. Therefore, it has been demonstrated that ontologies can be used in DBMI through effective knowledge extraction process assistance.

We do believe that, the integration of ontology engineering and KDD will play an important role in the semantic technologies adoption. Thus, this work contributes to both research and business applications by suggesting a hybrid methodology which describes the best practices and leverages both in ontology engineering and KDD processes.

5.4 Further work

This research is innovative and bridges different scientific domain areas: ontologies, technologies and information systems, and database marketing. This approach has brought to light a multidisciplinary approach which can be used in later research.

Some key areas of future work for the DBMI system prototype can be investigated along three related dimensions. In the first dimension, ontologies need to be evaluated with many other interactive and practical experiments. In the second dimension, the modular approach of KDD ontology must be furtherly developed in order to fully support all KDD related knowledge and not only the KDD process phases. Lastly in a third dimension, since both ontologies were developed in order to support a common overall DBM process, the ontologies integration should be investigated in order to expand their knowledge base, focusing also other marketing approaches.

Another key area regarding future work points to the user interface. Since ontologies promote the knowledge share and reuse, the development of ontology based systems that effectively assist end users in process development should be stressed.

However, for the computer scientist community this research addresses a bigger objective: a system that automatically generates the code which could be used to assure the entire process feedback loop. Thereafter, this code would be included into the knowledge base and then automatically reused in further interactions.

References

A

(Agrawal *et al.*, 1993) Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In on Management of Data, I. C., editor, *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216. International Conference on Management of Data.

(Anand *et al.*, 2007) Anand, S. S., Grobelnik, M., Herrmann, F., Hornick, M., Lingenfelder, C., Rooney, N., and Wettschereck, D. (2007). Knowledge discovery standards. *Artificial Intelligence Review*, 27(1):21–56.

(Ankerst *et al.*, 2003) Ankerst, M., Jones, D., Kao, A., and Wang, C. (2003). Datajewel: Tightly integrating visualization with temporal data mining. In *ICDM Workshop on Visual Data Mining*.

(Armstrong, 2006) Armstrong, J. S. (2006). Findings from evidence-based forecasting: Methods for reducing forecast error. *International Journal of Forecasting*, 22(3):583–598.

(Arndt and Gersten, 2001) Arndt, D. and Gersten, W. (2001). Data management in analytical customer relationship management. In *ECML/PKDD 2001 Workshop Proceedings*. DATA MINING FOR MARKETING APPLICATIONS.

(Arpirez *et al.*, 2000) Arpirez, J. C., Gomez-Perez, A., Lozano-Tello, A., and Pinto, H. S. A. (2000). Reference ontology and (onto)2 agent: The ontology yellow pages. *Knowledge and Information Systems*, 2(4):387–412.

B

(Baader *et al.*, 2003) Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2003). *The Descriprion Logic: Theory, Implementation and Applications*. Cambridge University Press.

(Baskerville, 1999) Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of AIS Volume 2, Article 19*, 2:19–51.

(Bean, 1999) Bean, R. (1999). Building a foundation for database marketing success. Technical report, DM Review Magazine.

(Bellandi *et al.*, 2006) Bellandi, A., Furletti, B., Grossi, V., and Romei, A. (2006). Ontology-driven association rule extraction: A case study. *Contexts and Ontologies: Representation and reasoning in 16th European Conference on Artificial Intelligence*, 1:1–10.

(Bernaras *et al.*, 1996) Bernaras, A., Laresgoiti, I., and Corera, J. M. (1996). Building and reusing ontologies for electrical network applications. In Wahlster, W., editor, *European Conference on Artificial Inteligence*, pages 298–302. 12th European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings, John Wiley and Sons, Chichester.

(Berners-Lee *et al.*, 2001) Berners-Lee, T., H., and J., Lassila, O. (2001). The semantic web. *Scientific American*, 284:34–43.

(Berners-Lee, 2003) Berners-Lee, T. (2003). Www past and future. Technical report, W3C.

(Bernstein *et al.*, 2005) Bernstein, A., Provost, F., and Hill, S. (2005). Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *IEEE Transactions on knowledge and data engineering*, 17(4).

(Berson and Smith, 2001) Berson, A. and Smith, S. J. (2001). *Data Warehousing, Data Mining &OLAP*. Computing Mcgraw-Hill, 3rd edition. ISBN 0-07-006272-2.

(Blanco *et al.*, 2008) Blanco, I. J., Vila, M. A., and Martinez-Cruz, C. (2008). The use of ontologies for representing database schemas of fuzzy information. *International Journal of intelligent Systems*, 23:419–445.

(Blazquez *et al.*, 1998) Blazquez, M., Fernandez, M., Garcia-Pinar, J. M., and Gomez-Perez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. In *Knowledge Acquisition Workshops and Archives*, Voyager Inn, Banff, Alberta, Canada. University of Calgary.

(Bohling *et al.*, 2006) Bohling, T., Bowman, D., LaValle, S., Mittal, V., Narayandas, D., Ramani, G., and Varadarajan, R. (2006). Crm implementation: Effectiveness issues and insights. *Journal of Service Research*, 9(2):184–194.

(Bolloju *et al.*, 2002) Bolloju, N., Khalifa, M., and Turban, E. (2002). Integrating knowledge management into enterprise environments for the next generation decision support. *Journal of Decision Support Systems*, 33(2):163–176.

(Bombardier *et al.*, 2007) Bombardier, V., Mazaud, C., Lhoste, P., and Vogrig, R. (2007). Contribution of fuzzy reasoning method to knowledge integration in a defect recognition system. *Journal of computers in industry*, 58(4):355–366.

(Bonnemaizon *et al.*, 2007) Bonnemaizon, A., Cova, B., and Louyot, M.-C. (2007). Relationship marketing in 2015: A delphi approach. *European Management Journal*, 25(1):50â€“59.

(Borges *et al.*, 2009) Borges, A. M., Corniel, M., Gil, R., Contreras, L., and Borges, R. (2009). Towards a study opportunities recommender system in ontological principles-based on semantic web environment. *WSEAS Transactions on Computers*, 8(2):279–291.

(Borst *et al.*, 1997) Borst, P., Akkermans, H., and Top, J. (1997). Engineering ontologies. *The International Journal of Human Computer Studies. In Special Issue: Using explicit ontologies in knowledge-based system development, Vol. 2/3 pp.365-406.*, 46(2-3):365–406.

(Bouquet *et al.*, 2002) Bouquet, P., Dona, A., Serafini, L., and Zanobini, S. (2002). Contextualized local ontology specification via ctxml. In for Artificial Intelligence, A. A., editor, *MeaN-02 AAAI Workshop on Meaning Negotiation*, Edmonton, Alberta, Canada. AAAI.

(Brazdil *et al.*, 2009) Brazdil, P., Giraud-Carrier, C., Soares, C., and Vilalta, R. (2009). Metalearning: Applications to data mining.

(Breiman *et al.*, 1984) Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. Wadsworth International Group.

(Brezany *et al.*, 2008) Brezany, P., Janciak, I., and Tjoa, A. M. (2008). *Data Mining with Ontologies: Implementations, Findings, and Frameworks*, chapter Ontology-Based Construction of Grid Data Mining Workflows, pages 182–210. Information Science Reference - IGI Global.

(Brito, 2000) Brito, P. Q. (2000). *Como Fazer Promocao de Vendas*. Mc Graw-Hill.

(Brito and Hammond, 2007) Brito, P. Q. and Hammond, K. (2007). Strategic versus tactical nature of sales promotions. *Journal of Marketing Communications*, 13(2):131–148.

(Brito *et al.*, 2004) Brito, P. Q., Jorge, A., and McGoldrick, P. J. (2004). The relationship between stores and shopping centers: Artificial intelligence and multivariate approach assessement. In *Proceedings of Annual Conference of IABE 2004*, Las Vegas, USA.

(Brookes *et al.*, 2004) Brookes, R. W., Brodie, R. J., Coviello, N. E., and Palmer, R. A. (2004). How managers perceive the impacts of information technologies on contemporary marketing practices: Reinforcing, enhancing or transforming? *Journal of Relationship Marketing*, 3(4):7–26.

(Buckinx and den Poel, 2005) Buckinx, W. and den Poel, D. V. (2005). Customer base analysis: Partial defection of behaviorally-loyal clients in a non-contractual fmcg retail setting. *European Journal of Operational Research*, 164 (1):252–268.

(Buckinx *et al.*, 2007) Buckinx, W., Verstraeten, G., and den Poel, D. V. (2007). Predicting customer loyalty using the internal transactional database. *Expert Systems with Applications*, 32:125–134.

(Burez and Poel, 2007) Burez, J. and Poel, D. V. (2007). Crm at a pay-tv company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications*, 32(2):277–288.

C

(Cannataro and Comito, 2003) Cannataro, M. and Comito, C. (2003). A data mining ontology for grid programming. In *First International Workshop on Semantics in Peer-to-Peer and Grid Computing, in conjunction with WWW2003*, pages 113–134.

(Cardoso and Lytras, 2009) Cardoso, J. and Lytras, M. (2009). *Semantic Web Engineering in the Knowledge Society*. Information Science Reference - IGI Global, New York.

(Carson *et al.*, 2004) Carson, D., Gilmore, A., and Walsh, S. (2004). Balancing transaction and relationship marketing in retail banking. *Journal of Marketing Management*, 20:431–455.

(Ceccaroni, 2001) Ceccaroni, L. (2001). *Ontoweeds - An ontology-based environmental decision-support system for the management of wastewater treatment plants*. PhD thesis, Universitat Politecnica de Catalunya, Barcelona.

(Cellini *et al.*, 2007) Cellini, J., Diamantini, C., and Potena, D. (2007). Kddbroker: Description and discovery of kdd services. In *15th Italian symposium on Advanced Database Systems*.

(CeSpivova *et al.*, 2004) CeSpivova, H., Rauch, J., Svatek, V., and Kejkula, M. (2004). Roles of medical ontology in association mining crisp-dm cycle. In *Knowledge Discovery and Ontologies*.

(Changchien and C., 2001) Changchien, S. and C., L. T. (2001). Mining association rules procedure to support on-line recommendation by customers and products fragmentation. *Expert Systems with Applications,*, 20(4):325â€“335.

(Cheng *et al.*, 2009) Cheng, H., Lu, Y.-C., and Sheu, C. (2009). Automated optimal equity portfolios discovery in a financial knowledge management system. *Expert Systems with Applications*, 36(2):3614–3622.

(Chu and Hwang, 2008)Chu, H.-C. and Hwang, G.-J. (2008). A delphi-based approach to developing expert systems with the cooperation of multiple experts. *Expert Systems with Applications*, 34:2826–2840.

(Cimiano *et al.*, 2004) Cimiano, P., Hotho, A., Stumme, G., and Tane, J. (2004). Conceptual knowledge processing with formal concept analysis and ontologies. In *ICFCA - Second International Conference on Formal Concept Analysis*.

(Cochran, 1983) Cochran, S. (1983). The delphi method: Formulation and refining group judgements. *Journal of Human Sciences*, 2(2):111–117.

(Corcho *et al.*, 2003) Corcho, O., Fernandez-Lopez, M., and Gomez-Perez, A. (2003). Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46:41–64.

(Coulet *et al.*, 2008) Coulet, A., Smail-Tabbone, M., Benlian, P., Napoli, A., and Devignes, M.-D. (2008). Ontology-guided data preparation for discovering genotype-phenotype relationships. *Journal of BMC Bioinformatics*, 9:1–9.

(Coviello and Brodie, 1998) Coviello, N. and Brodie, J. (1998). From transaction to relationship marketing: an investigation of managerial perceptions and practices. *Journal of Strategic Marketing*, 6(3):171–186.

(Coviello *et al.*, 2001) Coviello, N., Milley, R., and Marcolin, B. (2001). Understanding it-enabled interactivity in contemporary marketing. *Journal of Interactive Marketing*, 15(4):18–33.

(Coviello *et al.*, 2006) Coviello, N., Winklhofer, H., and Hamilton, K. (2006). Marketing practices and performance of small service firms - an examination in the tourism accommodation sector. *Journal of Service Research*, 9(1):38–58.

D

(Dabholkar and Neeley, 1998) Dabholkar, P. A. and Neeley, S. M. (1998). Managing interdependency: a taxonomy for business-to-business relationships. *Journal of Business and Industrial Marketing*, 13:439–460.

(Decker *et al.*, 1998) Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1998). Ontobroker: Ontology based access to distributed and semi-structured information. In *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher.

(Delbecq *et al.*, 1975) Delbecq, A. L., Ven, A. H. V. D., and Gustafson, D. H. (1975). *Group Techniques for Planning- A Guide to Nominal Group and Delphi Processes*. Scott.

(den Poel and Buckinx, 2005) den Poel, D. V. and Buckinx, W. (2005). Predicting online-purchasing behaviour. *European Journal of Operational Research Dirk Van den Poel and Wouter Buckinx*, 166(2):557–575.

(DeTienne and Thompson, 1996) DeTienne, K. B. and Thompson, J. A. (1996). Database marketing and organizational learning theory: toward a research agenda. *Journal of Consumer Marketing*, 13(5):12–34.

(Diamantini *et al.*, 2004) Diamantini, C., Panti, M., and Potena, D. (2004). Services for knowledge discovery in databases. In *In Int. Symposium of Santa Caterina on Challanges in the Internet and Interdisciplinary Research SSCCII-04*, volume 1.

(Diamantini *et al.*, 2006a) Diamantini, C., Potena, D., and Cellini, J. (2006a). Uddi registry for knowledge discovery in databases services. In *Proc. of AAAI Fall Symposium on Semantic Web for Collaborative Knowledge Acquisition*, pages 94–97, Arlington, VA, USA.

(Diamantini *et al.*, 2006b) Diamantini, C., Potena, D., and Smari, W. (2006b). Collaborative knowledge discovery in databases: A knowledge exchange perspective. In *Fall Symposium on Semantic Web for Collaborative Knowledge Acquisition*, pages 24–31, Arlington, VA, USA. AAAI, AAAI.

(Dick, 2008) Dick, B. (2008). Postgraduate programs using action research in action learning, action research and process management: Theory, practice, praxis. Technical report, Faculty of Education, Griffith University.

(Domingos, 2003) Domingos, P. (2003). Prospects and challenges for multi-relational data mining. *SIGKDD Explorer Newsletter*, 5(1):80–83.

(Domingues and Rezende, 2005) Domingues, M. A. and Rezende, S. O. (2005). Using taxonomies to facilitate the analysis of the association rules. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ; 2nd Knowledge Discovery and Ontolkogies Workshop*, volume 1, pages 59–66.

(Drozdenko and Perry, 2002) Drozdenko, R. and Perry, D. (2002). *Optimal Database Marketing*. SAGE Publications, Thousand Oaks, USA.

E

(Ekes *et al.*, 1997) Ekes, R., Farquhar, A., and Rice, J. (1997). Tools for assembling modular ontologies in ontolingua. In *AAAI-97 Proceedings*.

(El-Ansary, 2006) El-Ansary, A. I. (2006). Marketing strategy: taxonomy and frameworks. *European Business Review*, 18:266–293.

(Engelbach *et al.*, 2006)Engelbach, W., Hohn, R., Weichhardt, F., and Bohm, K. (2006). Ontology supported search engine and knowledge organisation, prototyped for international niche market information. *Proceedings of I-KNOW 2006*, 1:270–278.

(Euler and Scholz, 2004) Euler, T. and Scholz, M. (2004). Using ontologies in a kdd workbench. In *ECAI-2004 Workshop on Ontology Learning and Population*.

F

(Farquhar *et al.*, 1997) Farquhar, A., Fikes, R., and Rice, J. (1997). Tools for assembling modular ontologies in ontolingua. In *In Proceedings of Association for the Advancement of Artificial Intelligence 97*, pages 436–441. AAAI Press.

(Fayyad *et al.*, 1996) Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. In Magazine, A., editor, *AI Magazine*, volume 17, pages 37–54, Univ Calif Irvine, Dept Comp & Informat Sci, Irvine, Ca, 92717 Gte Labs Inc, Knowledge Discovery Databases Kdd Project, Tech Staff, Waltham, Ma, 02254. American Association for Artificial Intelligence.

(Fayyad and Uthurusamy, 1996) Fayyad, U. and Uthurusamy, R. (1996). Data mining and knowledge discovery in databases. In ACM, editor, *Communications of the ACM*, volume 39, pages 24–26, New York, NY, USA. ACM.

(Fensel *et al.*, 2000) Fensel, D., Horrocks, Harmelen, V., Decker, S., Erdmann, M., and Klein1, M. (2000). Oil in a nutshell. *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management - EKAW'00 Lecture Notes in Artificial Intelligence*, 1937:1–16.

(Fernandez *et al.*, 1997)Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: From ontological art towards ontological engineering. Technical report, AAAI.

(Fisher, 1987) Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Journal Machine Learning*, 2(2):139–172.

(Fletcher *et al.*, 1996) Fletcher, K., Wright, G., and Desai, C. (1996). The role of organizational factors in the adoption and sophistication of database marketing in the uk financial services industry. *Journal of Direct Marketing*, 10:10–21.

(Fox and Gruninger, 1997) Fox, M. and Gruninger, M. (1997). On ontologies and enterprise modelling. In Springer-Verlag, editor, *Proceedings of International Conference on Enterprise Integration Modelling Technology*. Springer- Verlag.

(Fox and Gruninger, 1998) Fox, M. S. and Gruninger, M. (1998). Enterprise modeling. *AAAI Magazine*, Fall1998:109–122.

(Frankland, 2007) Frankland, D. (2007). How firms use database marketing services. Technical report with user interview data, ForresterResearch Inc.

(Friedman-Hill and Scuse, 2008) Friedman-Hill, E. and Scuse, D. (2008). Jess: The rule engine for the java platform. Technical report, Sandia National Laboratories.

G

(Gersten *et al.*, 2000) Gersten, W., Wirth, R., and Arndt, D. (2000). Predictive modeling in automotive direct marketing: Tools, experiences and open issues. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining KDD '00*, pages 398 – 406. ACM New York, NY, USA.

(Giudici and Passerone, 2002) Giudici, P. and Passerone, G. (2002). Data mining of association structures to model consume rbehaviour. *Computational Statistics & Data Analysis*, 38:533â€“541.

(Gomez-Perez *et al.*, 2004) Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2004). *Ontological engineering*. Springer, 2nd edition.

(Gomez-Perez and Rojas-Amaya, 1999)Gomez-Perez, A. and Rojas-Amaya, D. (1999). Ontological reengineering for reuse. In *EKAW '99: Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 139–156, London, UK. Springer-Verlag.

(Gottgtroy *et al.*, 2004) Gottgtroy, P., Kasabov, N., and MacDonell, S. (2004). An ontology driven approach for knowledge discovery in biomedicine.

(Gronroos, 1994) Gronroos, C. (1994). From marketing mix to relationship marketing:towards a paradigm shift in marketing. *Management Decision*, 32(2):4–20.

(Gruber, 1993) Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220.

(Gruber *et al.*, 1990) Gruber, T. R., Pang, D., and Rice, J. (1990). Ontolingua: A language to support shared ontologies. Technical Report KSL-90-84, Knowledge Systems, AI Laboratory.

(Guarino, 1995)Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human and Computer Studies*, 43:625–640.

(Guarino, 1998)Guarino, N. (1998). Formal ontology and information systems. In Guarino, N., editor, *FOIS'98 Formal Ontology in Information systems*, pages 3–15, Amsterdam. IOS Press.

H

(Han and Kamber, 2001) Han, J. and Kamber, M. (2001). *Data mining: concepts and techniques*. Morgan Kaufman, San Francisco, CA.

(Honavar *et al.*, 2001) Honavar, V., Andorf, C., Caragea, D., Silvescu, A., Reinoso-Castillo, J., and Dobbs, D. (2001). Ontology-driven information extraction and knowledge acquisition from heterogeneous,distributed, autonomous biological data sources. In *Proceedings of the IJCAI-2001 Workshop on Knowledge Discovery from Heterogeneous, Distributed, Autonomous, Dynamic Data and Knowledge Sources.*

(Horrocks, 2003) Horrocks, I. (2003). Implementation and optimisation techniques. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 9, pages 306–346. Cambridge University Press.

(Horrocks *et al.*, 2005) Horrocks, I., Parsia, B., Patel-Schneider, P., and Hendler, J. (2005). Semantic web architecture: Stack or two towers? *Principles and Practice of Semantic Web Reasoning (PPSWR 2005)*, 3703(springer):37–41.

(Horrocks *et al.*, 2004) Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). Swrl: A semantic web rule language - combining owl and ruleml. Technical report, W3C.

(Hunyadi and Pah, 2008) Hunyadi, D. and Pah, I. (2008). Ontology used in a e-learning multi-agent architecture. *WSEAS Trans. Info. Sci. and App.*, 5(8):1302–1312.

J

(Jarrar, 2005) Jarrar, M. (2005). *Towards Methodological Principles for Ontology Engineering*. PhD thesis, Vrije Universiteit Brussel, Faculty of science.

(Jasper and Uschold, 1999) Jasper, R. and Uschold, M. (1999). A framework for understanding and classifying ontology applications. In *IJCAI 99 Ontology Workshop*, pages 16–21.

(Jones *et al.*, 1998) Jones, D., Bench-capon, T., and Visser, P. (1998). Methodologies for ontology development. In *Proceedings of Congress IT & KNOWS information technologies and knowledge systems*, Vienna, Austria. unknown.

(Jurisica *et al.*, 1999) Jurisica, I., Mylopoulos, J., and Yu, E. (1999). Ontologies for knowledge management: An information systems perspective. In for Information Sciences, A. S., editor, *Proceedings of the Annual Conference of the American Society for Information Sciences (ASISâ€™99)*. American Society for Information Sciences.

K

(Kalfoglou and Robertson, 2000) Kalfoglou, Y. and Robertson, D. (2000). Applying experienceware to support ontology deployment. In *In Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE00)*, pages 266–275.

(Kalfoglou and Schorlemmer, 2007) Kalfoglou, Y. and Schorlemmer, M. (2007). Ontology mapping: the state of the art. *Knowledge Engineering Review.*, 18(1):1–31.

(Kamber *et al.*, 1997) Kamber, M., Winstone, L., Gong, W., Cheng, S., and Han, J. (1997). Generalization and decision tree induction: efficient classification in data mining. In *RIDE '97: Proceedings of the 7th International Workshop on Research Issues in Data*

Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications, page 111, Washington, DC, USA. IEEE Computer Society.

(Kasabov *et al.*, 2007) Kasabov, N., Jain, V., Gottgtroy, P., Benuskova, L., Wysoski, S., and Joseph, F. (2007). Evolving brain-gene ontology system (ebgos): Towards integrating bioinformatics and neuroinformatics data to facilitate discoveries. In IEEE, editor, *International Joint Conference on Neural Networks*, Orlando - US.

(Kifer *et al.*, 1995) Kifer, M., Lausen, G., and Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843.

(Kim, 2008) Kim, Y. (2008). Boosting and measuring the performance of ensembles for a successful database marketing. *Expert Systems with Applications*, 1:17.

(Kishore *et al.*, 2004) Kishore, R., Zhang, H., and Ramesh, R. (2004). A helix-spindle model for ontological engineering. *Commun. ACM*, 47(2):69–75.

(Knublauch *et al.*, 2004a) Knublauch, H., Fergerson, R., Noy, N., and Musen., M. (2004a). The protege owl plugin: An open development environment for semantic web applications. In *The Semantic Web â€“ ISWC 2004*, pages 229–243. Springer.

(Knublauch *et al.*, 2004b) Knublauch, H., Musen, M. A., and Rector, A. L. (2004b). Editing description logic ontologies with the protege owl plugin. In *International Workshop on Description Logic - DL2004*.

(Kopanas *et al.*, 2002) Kopanas, I., Avouris, N. M., and Daskalaki, S. (2002). *The Role of Domain Knowledge in a Large Scale Data Mining Project*, volume 2308 of *Lecture Notes in Computer Science*, chapter Methods and Applications of Artificial Intelligence, pages 288–299. Springer Berlin / Heidelberg.

(Kotasek and Zendulka, 2000) Kotasek, P. and Zendulka, J. (2000). An xml framework proposal for knowledge discovery in databases. In *PKDD Workshop on Knowledge Management: Theory and Applications,*, volume 143-156. Machine Learning and Textual Information Access.

(Kuo *et al.*, 2007a) Kuo, R., Lin, S., and Shih, C. (2007a). Mining association rules through integration of clustering analysis and ant colony system for health insurance database in taiwan. *Expert Systems with Applications*, 33:794â€“808.

(Kuo *et al.*, 2007b) Kuo, Y.-T., Lonie, A., Sonenberg, L., and Paizis, K. (2007b). Domain ontology driven data mining: A medical case study. In on Domain Driven Data Mining DDDM2007, A. S. W., editor, *Conference on Knowledge Discovery in Data*, pages 11–17, California USA. ACM.

(Kurt *et al.*, 2008) Kurt, I., Ture, M., and Kurum, A. T. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. *Expert Systems with Applications*, 34(1):366–374.

(Kurtulus and Kurtulus, 2006) Kurtulus, S. and Kurtulus, K. (2006). Prospects, problems of marketing research and data mining in turkey. *Proceedings of World Academy of Science, Engineering and Technology*, 11:18–23. ISSN 1307-6884.

L

(Lariviere and den Poel, 2005) Lariviere, B. and den Poel, D. V. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29(2):472–484.

(Leary *et al.*, 2004) Leary, C. O., Rao, S., and Perry, C. (2004). Improving customer relationship management through database/internet marketing. *European Journal of Marketing*, 38(3/4):338–354.

(Lin and Hong, 2008) Lin, C. and Hong, C. (2008). Using customer knowledge in designing electronic catalog. *Expert systems with Applications*, 34:119–127.

(Linstone and Turoff, 2002) Linstone, H. A. and Turoff, M. (2002). *The Delphi Method - Techniques and Applications*. Murray Turoff and Harold A. Linstone.

(Lixiang, 2001) Lixiang, S. (2001). *Data mining thechniques based on rough set theory*. PhD thesis, National University of Singapore.

(Lomax, 2002) Lomax, P. (2002). *Research methods in educational leadership and management*, chapter Action Research, pages 122–140. Sage.

(Lopez, 1999) Lopez, M. F. (1999). Overview of methodologies for building ontologies. In V.R. Benjamins, B. Chandrasekaran, A. G.-P. N. G. M. U. e., editor, *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, volume 18.

(Lopez *et al.*, 1999) Lopez, M. F., Gomez-Perez, A., Sierra, J. P., and Sierra, A. P. (1999). Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems Journal*, 1:37–46.

(Lovrencic and Cubrilo, 2007) Lovrencic, S. and Cubrilo, M. (2007). University studies ontology - domain modeling. In *INES 2007 - International Conference on Intelligent Engineering Systems*.

M

(Madeira, 2002) Madeira, S. A. C. (2002). *Comparison of Target Selection Methods in Direct Marketing*. PhD thesis, Instituto TÃ©cnico - Universidade Tecnica de Lisboa.

(Marsh, 2005) Marsh, R. (2005). Drowning in dirty data. *Database Marketing & Customer Strategy Management*, 12(2):105–112.

(McClymont and Jocumsen, 2003) McClymont, H. and Jocumsen, G. (2003). How to implement marketing strategies using database approaches. *The Journal of Database Marketing & Customer Strategy Management*, 11(2):135–148.

(Merriam, 1998) Merriam, S. B. (1998). *Qualitative research and case study applications in education*. Jossey-Bass.

(Michalewicz *et al.*, 2006) Michalewicz, Z., Schmidt, M., Michalewicz, M., and Chiriac, C. (2006). *Adaptive Business Intelligence*. Springer.

(Motta, 1998) Motta, E. (1998). An overview of the ocml modelling language. In *In Proceedings KEML'98: 8th Workshop on Knowledge Engineering Methods & Languages*, pages 21–22.

(Murry and , 1995) Murry, J. and , J. H. (1995). Delphi: A versatile methodology for conducting qualitative research. *Review of Higher Education*, 18:423–436.

N

(Naik and Tsai, 2004) Naik, P. A. and Tsai, C.-L. (2004). Isotonic single-index model for high-dimensional database marketing. *Computational Statistics & Data Analysis 47 (2004) 775 â€“ 790*, 47:775–790.

(Neches *et al.*, 1991) Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *Artificial Intelligence Magazine*, 12(3):36–56.

(Nedellec and Nazarenko, 2005) Nedellec, C. and Nazarenko, A. (2005). Ontologies and information extraction. In *LIPN Internal Report*. LIPN.

(Newell and level, 1982) Newell, A. and level, T. (1982). The knowledge level. *Artificial Intelligence*, 18:87–127.

(NG and LIU, 2000) NG, K. and LIU, H. (2000). Customer retention via data mining. *Artificial Intelligence Review Issues on the Application of Data Mining.*, 14:569â€“590.

(Nigro *et al.*, 2008) Nigro, H. O., Cisaro, S. G., and Xodo, D. (2008). *Data Mining with Ontologies: Implementations, Findings and Frameworks*. Information Science Reference. Information Science Reference - IGI Global, London, igi global edition.

(Nogueira *et al.*, 2007) Nogueira, B. M., Santos, T. R. A., and ZÃ¡rate, L. E. (2007). Comparison of classifiers efficiency on missing values recovering: Application in a marketing database with massive missing data. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)*.

(Noy and McGuinness, 2003) Noy, N. F. and McGuinness, D. L. (2003). Ontology development 101: A guide to creating your first ontology. Technical report, Stanford University.

O

(O'Brien, 2002) O'Brien, R. (2002). *Theory and Practice of Action Research*, chapter An Overview of the Methodological Approach of Action Research. Universidade Federal da Paraiba.

(Ozimek, 2004) Ozimek, J. (2004). Case studies: The 2003 information management project awards. *Journal of Database Marketing & Customer Strategy Management*, 12(1):55.

P

(Parsia and Sirin, 2004) Parsia, B. and Sirin, E. (2004). Pellet: An owl dl reasoner. In *3rd International Semantic Web Conference (ISWC2004)*.

(Payne and Frow, 2005) Payne, A. and Frow, P. (2005). A strategic framework for customer relationship management. *Journal of Marketing*, 69(4):167–176.

(Pearce *et al.*, 2002) Pearce, J. E., Webb, G. I., Shaw, R. N., and Garner, B. (2002). A systemic approach to the database marketing process. *ANZMAC Conference Proceedings*, 1:2941–2948.

(Perez-Rey *et al.*, 2006) Perez-Rey, D., Anguita, A., and Crespo, J. (2006). *Biological and Medical Data Analysis*, volume 4345/2006 of *Lecture Notes in Computer Science*, chapter OntoDataClean: Ontology-Based Integration and Preprocessing of Distributed Data, pages 262–272. Springer Berlin / Heidelberg.

(Phillips and Buchanan, 2001) Phillips, J. and Buchanan, B. G. (2001). Ontology-guided knowledge discovery in databases. In ACM, editor, *International Conference On Knowledge Capture 1st international conference on Knowledge capture*, pages 123–130. International Conference On Knowledge Capture.

(Piatetsky-Shapiro, 1991) Piatetsky-Shapiro, G. (1991). Knowledge discovery in real databases: A workshop report. *AI Magazine*, 11:68–70.

(Piatetsky-Shapiro, 2007) Piatetsky-Shapiro, G. (2007). Data mining and knowledge discovery - 1996 to 2005: Overcoming the hype and moving from "university" to "business" and "analytics". *Data Mining and Knowledge Discovery journal*, 15:99–105.

(Pinto, 2006) Pinto, F. (2006). A descoberta de conhecimento em bases de dados como suporte a actividades de business intelligence - aplicaÃ§Ã£o na Ã rea do database marketing. Master's thesis, Universidade do Minho.

(Pinto *et al.*, 2009) Pinto, F., Santos, M. F., and Marques, A. (2009). *WSEAS Transactions on Business and Economics*, volume 6, chapter Database marketing intelligence supported by ontologies, pages 135–146. World Scientific and Engineering Academy and Society.

(Pinto and Martins, 2004) Pinto, H. S. and Martins, J. P. (2004). Ontologies: How can they be built? *Knowledge and Information Systems*, 6:441–464.

(Predoiu and Grimm, 2006) Predoiu, L. and Grimm, S. (2006). Ontology reasoning and querying - reasoner technology scan and recommendation. Technical report, DIP - Data, Information and Process Integration with Semantic Web Services.

(Prinzie and Poel, 2006) Prinzie, A. and Poel, D. V. (2006). Exploiting randomness for feature selection in multinomial logit: a crm cross-sell application. *Lecture Notes in Artificial Intelligence*, 4065:310–323.

Q

(Quine, 1992) Quine, W. V. (1992). *Theories and Things (Revised ed.)*. Harvard University Press.

(Quinlan, 1986)Quinlan, R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

R

(Rebelo *et al.*, 2006) Rebelo, C., Brito, P. Q., Soares, C., and Jorge, A. (2006). Factor analysis to support the visualization and interpretation of clusters of portal users. In *WI 2006: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 987–990, Washington, DC, USA. IEEE Computer Society.

(Rothenfluh *et al.*, 1996) Rothenfluh, T. E., Gennari, J. H., Eriksson, H., Puerta, A. R., Tu, S. W., and Musen, M. A. (1996). Reusable ontologies, knowledge-acquisition tools, and performance systems: Protege-ii solutionsto sisyphus-2. *International Journal of Human-Computer Studies 44: 303-332.*, 44:303–332.

(Rowe and Wright, 2001) Rowe, G. and Wright, G. (2001). Expert opinions in forecasting: The role of the delphi technique. *Principles of Forecasting*, pages 125–144. Kluwer Academic Publishers,.

(Rygielski *et al.*, 2002) Rygielski, C., Wang, J.-C., and Yen, D. C. (2002). Data mining techniques for customer relationship management. *Technology in Society*, 24:483–502.

S

(Santos *et al.*, 2005) Santos, M. F., Cortez, P., Quintela, H., and Pinto, F. (2005). A clustering approach for knowledge discovery in database marketing. *Data Mining VI: Data Mining, Text Mining and their Business Applications*, 35:399–407.

(Sarker *et al.*, 2002) Sarker, R., Abbass, H., and Newton, C. (2002). *Heuristics and Optimization for Knowledge Discovery*, chapter Introducing Data Mining and Knowledge Discovery. Idea Group Publishing.

(Schoenbachler *et al.*, 1997) Schoenbachler, D. D., Gordon, G. L., Foley, D., and Spellman, L. (1997). Understanding consumer database marketing. *Journal of Consumer Marketing*, 14(1):5–19.

(Seller and Gray, 1999) Seller, M. and Gray, P. (1999). A survey database marketing. Technical report, Center for Research on information Technology and Organizations.

(Sen and Tuzhiln, 1998) Sen, S. and Tuzhiln, A. (1998). Making sense of marketing data: Some mis perspectives on the analysis of large data sets. *Journal of Market Focused Management*, 3:91–111.

(Sharma and Osei-Bryson, 2008) Sharma, S. and Osei-Bryson, K.-M. (2008). Framework for formal implementation of the business understanding phase of data mining projects. *Expert Systems with Applications*, page in press.

(Shen and Chuang, 2009) Shen, C.-C. and Chuang, H.-M. (2009). A study on the applications of data mining techniques to enhance customer lifetime value. *WSEAS Transactions Information Science and Applications*, 6(2):319–328.

(Shepard, 1998)Shepard, D. (1998). *The New Direct Marketing: How to Implement A Profit-Driven Database Marketing Strategy*. David Shepard Ass, 3rd edition.

(Shi *et al.*, 2006) Shi, Y., Liu, J., Wang, R., and Chen, M. (2006). Developing methodologies of knowledge discovery and data mining to investigate metropolitan land use evolution. *PRICAI 2006: Trends in Artificial Intelligence*, 4099/2006:787 â€“ 796.

(Silva *et al.*, 2004) Silva, A., Cortez, P., Santos, M., Gomes, L., and Neves, J. (2004). Multiple organ failure diagnosis using adverse events and neural networks. In INSTICC, editor, *INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS*, volume 2, pages 401–408.

(Smith and Welty, 2001) Smith, B. and Welty, C. (2001). Ontology: Towards a new synthesis. In ACM, editor, *F0lS'01*, pages iii–ix. ACM.

(Smith and Farquhar, 2008) Smith, R. G. and Farquhar, A. (2008). The road ahead for knowledge management: An ai perspective. *American Association for Artificial Intelligence*, 1:17–40.

(Sowa, 2000) Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co.

(Staab and Studer, 2004) Staab, S. and Studer, R. (2004). *Handbook on ontologies*. International handbooks on information systems. Springer-Verlag.

(Swartout *et al.*, 1996) Swartout, B., Patil, R., Knight, K., and Russ, T. (1996). Ontosaurus: A tool for browsing and editing ontologies. *Knowledge Acquisition Workshops*, 1.

T

(Tao and Yeh, 2003) Tao, Y.-H. and Yeh, C.-C. R. (2003). Simple database marketing tools in customer analysis and retention. *International Journal of Information Management*, 23:291–301.

(Tsarkov and Horrocks, 2004) Tsarkov, D. and Horrocks, I. (2004). Reasoner demonstrator: Implementing new reasoner with datatypes support. Technical report, University of Manchester.

(Tsarkov and Horrocks, 2006) Tsarkov, D. and Horrocks, I. (2006). Fact++ description logic reasoner: System description. In springer, editor, *Int. Joint Conferene on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. springer.

(Tudorache, 2006) Tudorache, T. (2006). *Employing Ontologies for an Improved Development Process in Collaborative Engineering*. PhD thesis, University of Berlim, Germany.

U

(Udrea *et al.*, 2007) Udrea, O., Getoor, L., and Miller, R. J. (2007). Leveraging data and structure in ontology integration. In of the 2007 ACM SIGMOD international conference on Management of data, P., editor, *International Conference on Management of Data*, pages 449–460.

(Uschold and Gruninger, 2004) Uschold, M. and Gruninger, M. (2004). Ontologies and semantics for seamless connectivity. *SIGMOD Rec.*, 33(4):58–64.

(Uschold and King, 1995) Uschold, M. and King, M. (1995). Towards a metthodology for building ontologies. *AIAI-TR Workshop on Basic Ontological Issues in Knowledge Sharing*, 1.

V

(van Heijst *et al.*, 1997) van Heijst, G., Schreiber, A. T., and Wielinga, B. J. (1997). Using explicit ontologies in kbs development. *International Journal of Human-Computer Studies*, 46(2):183–292.

(Verhoef and Hoekstra, 1999) Verhoef, P. and Hoekstra, J. (1999). Status of database marketing in the dutch fast moving consumer goods industry. *Journal of Market Focused Management*, 3:313â€“331.

(Vernette, 1997) Vernette, E. (1997). Evaluation de la validation predictive de la methode delphi-leader. In *Congres International de l AFM*, page 988 1010.

(Vilalta *et al.*, 2005) Vilalta, R., Giraud-Carrier, C., and Brazdil, P. (2005). Meta-learning - concepts and techniques. In Maimon O, R. L., editor, *in Data Mining and Knowledge Discovery Handbook*, pages 731–748. Springer. ISIProc, DBLP.

(Vindevogel *et al.*, 2005) Vindevogel, B., Poel, D. V., and Wets, G. (2005). Why promotion strategies based on market basket analysis do not work. *Expert Systems with Applications*, 28(3):583–590.

W

(Wehmeyer, 2005) Wehmeyer, K. (2005). Aligning it and marketing - the impact of database marketing and crm. *Journal of Database Marketing & Customer Strategy Management*, 12(2):243.

(Welty and Murdock, 2006) Welty, C. and Murdock, J. W. (2006). Towards knowledge acquisition from information extraction. In Springer, editor, *In Proceedings of ISWC-2006.*, Athens.

(Weng and Chang, 2008) Weng, S.-S. and Chang, H.-L. (2008). Using ontologies network analysis for research document recommendation. *Expert Systems with Applications*, 34:1857–1869.

(Witten and Frank, 2000) Witten, I. H. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Technique*. The Morgan Kaufmann Series in Data Management Systems, 2nd edition.

Y

(Yohannes and Hoddinott, 1999) Yohannes, Y. and Hoddinott, J. (1999). *Classification and Regression Trees: An Introduction*. International Food Policy Research Institute.

Z

(Zairate *et al.*, 2006) Zairate, L. E., Nogueira, B. M., Santos, T. R. A., and Song, M. A. J. (2006). Techniques for missing value recovering in imbalanced databases: Application in a marketing database with massive missing data. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2658–2664, Taiwan. IEEE.

(Zhou *et al.*, 2006) Zhou, X., Geller, J., and Halper, Y. P. M. (2006). *An Application Intersection Marketing Ontology*, chapter Theoretical Computer Science, pages 143–163. Lecture Notes in Computer Science. Springer Berlin / Heidelberg.

(Zineldin and Vasicheva, 2008) Zineldin, M. and Vasicheva, V. (2008). Cybernization management in the cyber world: a new management perspective. *Problems and Perspectives in Management, Volume 6, Issue 1, 2008*, 1:113–126.

(Zubber-Skerritt, 2000) Zubber-Skerritt, O. (2000). *New Directions in Action Research*. Falmer Press.

(Zuber-Skerrit and Perry, 2000) Zuber-Skerrit and Perry, C. (2000). Action research in graduate management theses. *Action Learning, Action Research and Process Management: Theory, Practice, Praxis*, 1:84.

(Zwick and Dholakia, 2004) Zwick, D. and Dholakia, N. (2004). Whose identity is it anyway? consumer representation in the age of database marketing. *Journal of Macromarketing*, 24(1):31–43.

Appendices

Appendix 1 - SWRL Built-Ins

This document contains a proposal for a Semantic Web Rule Language (SWRL) Built Ins[25] based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language. The proposal extends the set of OWL axioms. It thus enables rules to be combined with an OWL knowledge base. A high-level abstract syntax is provided that extends the OWL abstract syntax described in the OWL Semantics and Abstract Syntax document.

The proposed rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: *whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold*.

Both the antecedent (body) and consequent (head) consist of zero or more atoms. An empty antecedent is treated as trivially true (i.e. satisfied by every interpretation), so the consequent must also be satisfied by every interpretation; an empty consequent is treated as trivially false (i.e., not satisfied by any interpretation), so the antecedent must also not be satisfied by any interpretation. Multiple atoms are treated as a conjunction. Note that rules with conjunctive consequents could easily be transformed (via the Lloyd-Topor transformations [Lloyd87]) into multiple rules each with an atomic consequent.

The set of built-ins for SWRL is motivated by a modular approach that will allow further extensions in future releases within a (hierarchical) taxonomy. At the same time, it will provide the flexibility for various implementations to select the modules to be supported with each version of SWRL.

This system of built-ins should also help in the interoperation of SWRL with other Web formalisms by providing an extensible, modular built-ins infrastructure for Semantic Web Languages, Web Services, and Web applications.

SWRL built-ins are used in builtin atoms. For example, *swrlx:builtinAtom* identifies a built-in using the *swrlx:builtin* attribute and lists its arguments as sub-elements.

[25] Swrl built-ins are identified using the http://www.w3.org/2003/11/swrlb namespace.

1. Built-Ins for Comparisons

swrlb:equal (from XQuery op:numeric-equal, op:compare, op:boolean-equal op:yearMonthDuration-equal, op:dayTimeDuration-equal, op:dateTime-equal, op:date-equal, op:time-equal, op:gYearMonth-equal, op:gYear-equal, op:gMonthDay-equal, op:gMonth-equal, op:gDay-equal, op:anyURI-equal) - Satisfied if the first argument and the second argument are the same.

swrlb:notEqual (from swrlb:equal) - The negation of swrlb:equal.

swrlb:lessThan (from XQuery op:numeric-less-than, op:compare, op:yearMonthDuration-less-than, op:dayTimeDuration-less-than, op:dateTime-less-than, op:date-less-than, op:time-less-than) - Satisfied iff the first argument and the second argument are both in some implemented type and the first argument is less than the second argument according to a type-specific ordering (partial or total), if there is one defined for the type. The ordering function for the type of untyped literals is the partial order defined as string ordering when the language tags are the same (or both missing) and incomparable otherwise.

swrlb:lessThanOrEqual (from swrlb:lessThan, swrlb:equal) - Either less than, as above, or equal, as above.

swrlb:greaterThan (from XQuery op:numeric-greater-than, op:compare, op:yearMonthDuration-greater-than, op:dayTimeDuration-greater-than, op:dateTime-greater-than, op:date-greater-than, op:time-greater-than) - Similarly to *swrlb:lessThan*;

swrlb:greaterThanOrEqual (from swrlb:greaterThan, swrlb:equal) - Similarly to *swrlb:lessThanOrEqual*;

2. Math Built-Ins

The following built-ins are defined for various numeric types. For the relation to be satisfied the arguments all have to belong to some numeric type for which the relation is defined;

swrlb:add (from XQuery op:numeric-add) - Satisfied if the first argument is equal to the arithmetic sum of the second argument through the last argument;

swrlb:subtract (from XQuery op:numeric-subtract) - Satisfied if the first argument is equal to the arithmetic difference of the second argument minus the third argument;

swrlb:multiply (from XQuery op:numeric-multiply) - Satisfied if the first argument is equal to the arithmetic product of the second argument through the last argument;

swrlb:divide (from XQuery op:numeric-divide) - Satisfied if the first argument is equal to the arithmetic quotient of the second argument divided by the third argument.

swrlb:integerDivide (from XQuery op:numeric-integer-divide) - Satisfied if the first argument is the arithmetic quotient of the second argument *idiv* the third argument. If the numerator is not evenly divided by the divisor, then the quotient is the *xsd:integer* value obtained, ignoring any remainder that results from the division (that is, no rounding is performed).

swrlb:mod (from XQuery op:numeric-mod) - Satisfied if the first argument represents the remainder resulting from dividing the second argument, the dividend, by the third argument, the divisor. The operation a mod b for operands that are *xsd:integer* or *xsd:decimal*, or types derived from them, produces a result such that *(a idiv b)*b+(a mod b) is equal to* a and the magnitude of the result is always less than the magnitude of *b*. This identity holds even in the special case that the dividend is the negative integer of largest possible magnitude for its type and the divisor is *-1 (the remainder is 0)*. It follows from this rule that the sign of the result is the sign of the dividend

swrlb:pow - Satisfied iff the first argument is equal to the result of the second argument raised to the third argument power;

swrlb:unaryPlus (from XQuery op:numeric-unary-plus) - Satisfied if the first argument is equal to the second argument with its sign unchanged;

swrlb:unaryMinus (from XQuery op:numeric-unary-minus) - Satisfied if the first argument is equal to the second argument with its sign reversed;

swrlb:abs (from XQuery fn:abs) - Satisfied if the first argument is the absolute value of the second argument;

swrlb:ceiling (from XQuery fn:ceiling) - Satisfied if the first argument is the smallest number with no fractional part that is greater than or equal to the second argument;

swrlb:floor (from XQuery fn:floor) - Satisfied if the first argument is the largest number with no fractional part that is less than or equal to the second argument;

swrlb:round (from XQuery fn:round) - Satisfied if the first argument is equal to the nearest number to the second argument with no fractional part;

swrlb:roundHalfToEven (from XQuery fn:round-half-to-even) - Satisfied if the first argument is equal to the second argument rounded to the given precision. If the fractional part is exactly half, the result is the number whose least significant digit is even;

swrlb:sin - Satisfied if the first argument is equal to the sine of the radian value the second argument;

swrlb:cos - Satisfied if the first argument is equal to the cosine of the radian value the second argument;

swrlb:tan - Satisfied if the first argument is equal to the tangent of the radian value the second argument;

3. Built-Ins for Boolean Values

swrlb:booleanNot (from XQuery fn:not) - Satisfied if the first argument is true and the second argument is false, or vice versa.

4. Built-Ins for Strings

The following built-ins are defined for strings (only), i.e., not untyped literals with language tags.

swrlb:stringEqualIgnoreCase - Satisfied iff the first argument is the same as the second argument (upper/lower case ignored)

swrlb:stringConcat (from XQuery fn:concat) - Satisfied if the first argument is equal to the string resulting from the concatenation of the strings the second argument through the last argument;

swrlb:substring (from XQuery fn:substring) - Satisfied iff the first argument is equal to the substring of optional length the fourth argument starting at character offset the third argument in the string the second argument;

swrlb:stringLength (from XQuery fn:string-length) - Satisfied if the first argument is equal to the length of the second argument;

swrlb:normalizeSpace (from XQuery fn:normalize-space) - Satisfied if the first argument is equal to the whitespace-normalized value of the second argument;

swrlb:upperCase (from XQuery fn:upper-case) - Satisfied iff the first argument is equal to the upper-cased value of the second argument;

swrlb:lowerCase (from XQuery fn:lower-case) - Satisfied iff the first argument is equal to the lower-cased value of the second argument;

swrlb:translate (from XQuery fn:translate) - Satisfied iff the first argument is equal to the second argument with occurrences of characters contained in the third argument replaced by the character at the corresponding position in the string the fourth argument;

swrlb:contains (from XQuery fn:contains) - Satisfied iff the first argument contains the second argument (case sensitive);

swrlb:containsIgnoreCase - Satisfied iff the first argument contains the second argument (case ignored);

swrlb:startsWith (from XQuery fn:starts-with) - Satisfied iff the first argument starts with the second argument-

swrlb:endsWith (from XQuery fn:ends-with) - Satisfied iff the first argument ends with the second argument.

swrlb:substringBefore (from XQuery fn:substring-before) - Satisfied iff the first argument is the characters of the second argument that precede the characters of the third argument.

swrlb:substringAfter (from XQuery fn:substring-after) - Satisfied iff the first argument is the characters of the second argument that follow the characters of the third argument.

swrlb:matches (from XQuery fn:matches) - Satisfied iff the first argument matches the regular expression the second argument.

swrlb:replace (from XQuery fn:replace) - Satisfied iff the first argument is equal to the value of the second argument with every substring matched by the regular expression the third argument replaced by the replacement string the fourth argument.

swrlb:tokenize (from XQuery fn:tokenize) - Satisfied iff the first argument is a sequence of one or more strings whose values are substrings of the second argument separated by substrings that match the regular expression the third argument.

5. Built-Ins for Date, Time and Duration

The following built-ins are defined for the XML Schema date, time, and duration datatypes, only, as appropriate.

swrlb:yearMonthDuration (from XQuery xdt:yearMonthDuration) - Satisfied iff the first argument is the xsd:duration representation consisting of the year the second argument and month the third argument.

swrlb:dayTimeDuration (from XQuery xdt:dayTimeDuration) - Satisfied iff the first argument is the xsd:duration representation consisting of the days the second argument, hours the third argument, minutes the fourth argument, and seconds the fifth argument.

swrlb:dateTime - Satisfied iff the first argument is the xsd:dateTime representation consisting of the year the second argument, month the third argument, day the fourth argument, hours the fifth argument, minutes the sixth argument, seconds the seventh argument, and timezone the eighth argument.

swrlb:date - Satisfied iff the first argument is the xsd:date representation consisting of the year the second argument, month the third argument, day the fourth argument, and timezone the fifth argument.

swrlb:time - Satisfied iff the first argument is the xsd:time representation consisting of the hours the second argument, minutes the third argument, seconds the fourth argument, and timezone the fifth argument.

swrlb:addYearMonthDurations (from XQuery op:add-yearMonthDurations)

Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic sum of the yearMonthDuration the second argument through the yearMonthDuration the last argument.

swrlb:subtractYearMonthDurations (from XQuery op:subtract-yearMonthDurations) - Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic difference of the yearMonthDuration the second argument minus the yearMonthDuration the third argument.

swrlb:multiplyYearMonthDuration (from XQuery op:multiply-yearMonthDuration) - Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic product of the yearMonthDuration the second argument multiplied by the third argument.

swrlb:divideYearMonthDuration (from XQuery op:divide-yearMonthDuration) - Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic remainder of the yearMonthDuration the second argument divided by the third argument.

swrlb:addDayTimeDurations (from XQuery op:add-dayTimeDurations) - Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic sum of the dayTimeDuration the second argument through the dayTimeDuration the last argument.

swrlb:subtractDayTimeDurations (from XQuery op:subtract-dayTimeDurations) - Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the dayTimeDuration the second argument minus the dayTimeDuration the third argument.

*swrlb:multiplyDayTimeDuration (*from XQuery op:multiply-dayTimeDuration) - Satisfied iff the *dayTimeDuration* the first argument is equal to the arithmetic product of the *dayTimeDuration* the second argument multiplied by the third argument;

swrlb:divideDayTimeDuration (from XQuery op:divide-dayTimeDuration) - Satisfied iff the *dayTimeDuration* the first argument is equal to the arithmetic remainder of the *dayTimeDuration* the second argument divided by the third argument;

swrlb:subtractDates (from XQuery op:subtract-dates) - Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the xsd:date the third argument;

swrlb:subtractTimes (from XQuery op:subtract-times) - Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:time the second argument minus the xsd:time the third argument;

swrlb:addYearMonthDurationToDateTime (from XQuery op:add-yearMonthDuration-to-dateTime) - Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic sum of the xsd:dateTime the second argument plus the *yearMonthDuration* the third argument;

swrlb:addDayTimeDurationToDateTime (from XQuery op:add-dayTimeDuration-to-dateTime) - Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic sum of the xsd:dateTime the second argument plus the *dayTimeDuration* the third argument;

swrlb:subtractYearMonthDurationFromDateTime (from XQuery op:subtract-yearMonthDuration-from-dateTime) - Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the *yearMonthDuration* the third argument;

swrlb:subtractDayTimeDurationFromDateTime (from XQuery op:subtract-dayTimeDuration-from-dateTime) - Satisfied iff the xsd:dateTime the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the dayTimeDuration the third argument;

swrlb:addYearMonthDurationToDate (from XQuery op:add-yearMonthDuration-to-date) - Satisfied iff the xsd:date the first argument is equal to the arithmetic sum of the xsd:date the second argument plus the *yearMonthDuration* the third argument;

swrlb:addDayTimeDurationToDate (from XQuery op:add-dayTimeDuration-to-date) - Satisfied iff the xsd:date the first argument is equal to the arithmetic sum of the xsd:date the second argument plus the dayTimeDuration the third argument;

swrlb:subtractYearMonthDurationFromDate (from XQuery op:subtract-yearMonthDuration-from-date) - Satisfied iff the xsd:date the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the yearMonthDuration the third argument;

swrlb:subtractDayTimeDurationFromDate (from XQuery op:subtract-dayTimeDuration-from-date) - Satisfied iff the xsd:date the first argument is equal to the arithmetic difference of the xsd:date the second argument minus the yearMonthDuration the third argument;

swrlb:addDayTimeDurationToTime (from XQuery op:add-dayTimeDuration-to-time)
Satisfied iff the xsd:time the first argument is equal to the arithmetic sum of the xsd:time the second argument plus the dayTimeDuration the third argument;

swrlb:subtractDayTimeDurationFromTime (from XQuery op:subtract-dayTimeDuration-from-time) - Satisfied iff the xsd:time the first argument is equal to the arithmetic difference of the xsd:time the second argument minus the dayTimeDuration the third argument;

swrlb:subtractDateTimesYieldingYearMonthDuration (from XQuery fn:subtract-dateTimes-yielding-yearMonthDuration) - Satisfied iff the yearMonthDuration the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the xsd:dateTime the third argument;

swrlb:subtractDateTimesYieldingDayTimeDuration (from XQuery fn:subtract-dateTimes-yielding-dayTimeDuration) - Satisfied iff the dayTimeDuration the first argument is equal to the arithmetic difference of the xsd:dateTime the second argument minus the xsd:dateTime the third argument;

6. Built-Ins for URIs

The following built-ins are defined for the XML Schema datatypes related to URIs.

swrlb:resolveURI (from XQuery op:resolve-uri) - Satisfied iff the URI reference the first argument is equal to the value of the URI reference the second argument resolved relative to the base URI the third argument.

swrlb:anyURI - Satisfied iff the first argument is a URI reference consisting of the scheme the second argument, host the third argument, port the fourth argument, path the fifth argument, query the sixth argument, and fragment the seventh argument.

7. Built-Ins for Lists

The following built-ins are defined for RDF-style lists. (Note that these built-ins are not usable in OWL DL or OWL Lite as RDF-style lists can only be used as OWL data in OWL Full.)

swrlb:listConcat (from Common Lisp append) - Satisfied iff the first argument is a list representing the concatenation of the lists the second argument through the last argument.

swrlb:listIntersection - Satisfied iff the first argument is a list containing elements found in both the list the second argument and the list the third argument.

swrlb:listSubtraction - Satisfied iff the first argument is a list containing the elements of the list the second argument that are not members of the list the third argument.

swrlb:member - Satisfied iff the first argument is a member of the list the second argument;

swrlb:length (from Common Lisp list-length) - Satisfied iff the first argument is the length of the list the second argument (number of members of the list);

swrlb:first (from rdf:first) - Satisfied iff the first argument is the first member of the list the second argument;

swrlb:rest (from rdf:rest) - Satisfied iff the first argument is a list containing all members of the list the second argument except the first member (the head);

swrlb:sublist - Satisfied iff the list the first argument contains the list the second argument;

swrlb:empty (from rdf:nil) - Satisfied iff the list the first argument is an empty list

Appendix 2 - KDD Ontology

General class hierarchy taxonomy tree.

Appendix 3 - KDD ontology OWL code.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns="http://www.owl-ontologies.com/Ontology1243503255.owl#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1243503255.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="DataTransform">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="DataPreProcessing"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="TestingData">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="ModelWorkingData"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Set">
   <owl:disjointWith>
    <owl:Class rdf:ID="Date"/>
   </owl:disjointWith>
   <owl:disjointWith>
    <owl:Class rdf:ID="Number"/>
   </owl:disjointWith>
   <owl:disjointWith>
    <owl:Class rdf:ID="Integer"/>
   </owl:disjointWith>
   <owl:disjointWith>
    <owl:Class rdf:ID="String"/>
   </owl:disjointWith>
   <rdfs:subClassOf>
    <owl:Class rdf:ID="StructureType"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DataDescription">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="DataUnderstand"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Recall">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="AlgorithmParameter"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ConsumerEvent">
   <rdfs:subClassOf>
```

```
    <owl:Class rdf:ID="Trigger"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ProcessPhase"/>
<owl:Class rdf:ID="DataPreparation">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Modeling"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="AttributeDerivation">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasAttributeDerivationTask"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:intersectionOf rdf:parseType="Collection">
            <owl:Class rdf:ID="ClienteData"/>
            <owl:Class rdf:about="#Date"/>
          </owl:intersectionOf>
        </owl:Class>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:intersectionOf rdf:parseType="Collection">
            <owl:Class rdf:ID="ProspectData"/>
            <owl:Class rdf:about="#Date"/>
          </owl:intersectionOf>
        </owl:Class>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasAttributeDerivationTask"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#DataPreProcessing"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="AlgorithmType">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Algorithm"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Other">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Trigger"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Demographics">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Personal"/>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:about="#String">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#StructureType"/>
 </rdfs:subClassOf>
 <owl:disjointWith>
  <owl:Class rdf:about="#Date"/>
 </owl:disjointWith>
 <owl:disjointWith>
  <owl:Class rdf:about="#Number"/>
 </owl:disjointWith>
 <owl:disjointWith>
  <owl:Class rdf:about="#Integer"/>
 </owl:disjointWith>
 <owl:disjointWith rdf:resource="#Set"/>
</owl:Class>
<owl:Class rdf:ID="DataSelection">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#DataUnderstand"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ObjectiveType">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#Modeling"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Society">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#Trigger"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TrainingData">
 <rdfs:subClassOf>
  <owl:Class rdf:about="#ModelWorkingData"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Modeling">
 <rdfs:subClassOf rdf:resource="#ProcessPhase"/>
</owl:Class>
<owl:Class rdf:about="#Personal">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="InformationType"/>
 </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Categorizer">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty>
    <owl:ObjectProperty rdf:ID="hasCategorizationTask"/>
   </owl:onProperty>
   <owl:someValuesFrom>
    <owl:Class>
     <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Number"/>
      <owl:Class rdf:about="#String"/>
     </owl:unionOf>
    </owl:Class>
   </owl:someValuesFrom>
```

```
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#DataPreProcessing"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Tree">
  <rdfs:subClassOf rdf:resource="#AlgorithmType"/>
</owl:Class>
<owl:Class rdf:ID="PersonalEvent">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Trigger"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GainRatio">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#AlgorithmParameter"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#DataUnderstand">
  <rdfs:subClassOf rdf:resource="#ProcessPhase"/>
</owl:Class>
<owl:Class rdf:about="#ClienteData">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasData"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Demographics"/>
            <owl:Class rdf:ID="Psychographics"/>
            <owl:Class rdf:ID="Transactional"/>
            <owl:Class rdf:ID="LifeStyle"/>
          </owl:unionOf>
        </owl:Class>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Data"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ModelWorkingData">
  <rdfs:subClassOf rdf:resource="#DataPreparation"/>
</owl:Class>
<owl:Class rdf:ID="ResultModel"/>
<owl:Class rdf:ID="Economic">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Environment"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Classifier">
  <rdfs:subClassOf rdf:resource="#AlgorithmType"/>
</owl:Class>
<owl:Class rdf:about="#AlgorithmParameter">
  <rdfs:subClassOf>
```

```
    <owl:Class rdf:about="#Algorithm"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Description">
  <rdfs:subClassOf rdf:resource="#ObjectiveType"/>
</owl:Class>
<owl:Class rdf:ID="Clusterer">
  <rdfs:subClassOf rdf:resource="#AlgorithmType"/>
</owl:Class>
<owl:Class rdf:ID="Resources"/>
<owl:Class rdf:about="#DataPreProcessing">
  <rdfs:subClassOf rdf:resource="#ProcessPhase"/>
</owl:Class>
<owl:Class rdf:about="#Data">
  <rdfs:subClassOf rdf:resource="#Resources"/>
</owl:Class>
<owl:Class rdf:about="#Trigger">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#InformationType"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Psychographics">
  <rdfs:subClassOf rdf:resource="#Personal"/>
</owl:Class>
<owl:Class rdf:about="#Algorithm">
  <rdfs:subClassOf rdf:resource="#Resources"/>
</owl:Class>
<owl:Class rdf:ID="DataTranslation">
  <rdfs:subClassOf rdf:resource="#DataUnderstand"/>
</owl:Class>
<owl:Class rdf:about="#LifeStyle">
  <rdfs:subClassOf rdf:resource="#Personal"/>
</owl:Class>
<owl:Class rdf:ID="Evaluation">
  <rdfs:subClassOf rdf:resource="#ProcessPhase"/>
</owl:Class>
<owl:Class rdf:about="#Integer">
  <owl:disjointWith>
    <owl:Class rdf:about="#Date"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Number"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#String"/>
  <owl:disjointWith rdf:resource="#Set"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StructureType"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Number">
  <owl:disjointWith>
    <owl:Class rdf:about="#Date"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Integer"/>
  <owl:disjointWith rdf:resource="#String"/>
  <owl:disjointWith rdf:resource="#Set"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#StructureType"/>
```

```
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="TestOption">
  <rdfs:subClassOf rdf:resource="#AlgorithmParameter"/>
 </owl:Class>
 <owl:Class rdf:ID="Outlier">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:someValuesFrom>
     <owl:Class rdf:about="#InformationType"/>
    </owl:someValuesFrom>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="hasOutlierTask"/>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#DataPreProcessing"/>
 </owl:Class>
 <owl:Class rdf:about="#Environment">
  <rdfs:subClassOf>
   <owl:Class rdf:about="#InformationType"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="Iteraction">
  <rdfs:subClassOf rdf:resource="#AlgorithmParameter"/>
 </owl:Class>
 <owl:Class rdf:ID="CondifenceFactor">
  <rdfs:subClassOf rdf:resource="#AlgorithmParameter"/>
 </owl:Class>
 <owl:Class rdf:about="#StructureType">
  <rdfs:subClassOf rdf:resource="#Data"/>
 </owl:Class>
 <owl:Class rdf:ID="Balance">
  <rdfs:subClassOf rdf:resource="#DataPreProcessing"/>
 </owl:Class>
 <owl:Class rdf:ID="AttributeDescription">
  <rdfs:subClassOf rdf:resource="#DataUnderstand"/>
 </owl:Class>
 <owl:Class rdf:about="#Date">
  <owl:disjointWith rdf:resource="#Number"/>
  <owl:disjointWith rdf:resource="#Integer"/>
  <owl:disjointWith rdf:resource="#String"/>
  <owl:disjointWith rdf:resource="#Set"/>
  <rdfs:subClassOf rdf:resource="#StructureType"/>
 </owl:Class>
 <owl:Class rdf:ID="Source">
  <rdfs:subClassOf rdf:resource="#Data"/>
 </owl:Class>
 <owl:Class rdf:ID="MissingValue">
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="hasMissingValueTask"/>
    </owl:onProperty>
    <owl:someValuesFrom>
     <owl:Class rdf:about="#InformationType"/>
    </owl:someValuesFrom>
   </owl:Restriction>
```

```
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#DataPreProcessing"/>
</owl:Class>
<owl:Class rdf:ID="Internal">
  <rdfs:subClassOf rdf:resource="#Source"/>
</owl:Class>
<owl:Class rdf:ID="Financial">
  <rdfs:subClassOf rdf:resource="#Environment"/>
</owl:Class>
<owl:Class rdf:ID="External">
  <rdfs:subClassOf rdf:resource="#Source"/>
</owl:Class>
<owl:Class rdf:ID="Social">
  <rdfs:subClassOf rdf:resource="#Environment"/>
</owl:Class>
<owl:Class rdf:ID="AlgorithmSelection">
  <rdfs:subClassOf rdf:resource="#Modeling"/>
</owl:Class>
<owl:Class rdf:ID="AttributeEvaluate">
  <rdfs:subClassOf rdf:resource="#DataPreparation"/>
</owl:Class>
<owl:Class rdf:about="#InformationType">
  <rdfs:subClassOf rdf:resource="#Data"/>
</owl:Class>
<owl:Class rdf:about="#Transactional">
  <rdfs:subClassOf rdf:resource="#Personal"/>
</owl:Class>
<owl:Class rdf:ID="Prediction">
  <rdfs:subClassOf rdf:resource="#ObjectiveType"/>
</owl:Class>
<owl:Class rdf:about="#ProspectData">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Demographics"/>
            <owl:Class rdf:about="#LifeStyle"/>
            <owl:Class rdf:about="#Psychographics"/>
          </owl:unionOf>
        </owl:Class>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasData"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Data"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasBalanceTask">
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="hasDataPreprocessing"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Balance"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataUnderstand">
  <rdfs:range rdf:resource="#DataUnderstand"/>
```

```
  <rdfs:domain rdf:resource="#ProcessPhase"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasTestingSet">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#DataPreparation"/>
        <owl:Class rdf:about="#TestingData"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:domain rdf:resource="#ModelWorkingData"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="hasDataPreparation"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEvaluation">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="hasAlgorithm"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasModelingObjective">
  <rdfs:range rdf:resource="#ObjectiveType"/>
  <rdfs:domain rdf:resource="#Modeling"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="hasModeling"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasModelingDataSelection">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasDataPreparation"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#DataPreparation"/>
  <rdfs:range rdf:resource="#ModelWorkingData"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPersonalEvent">
  <rdfs:domain rdf:resource="#Trigger"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#InformationType"/>
        <owl:Class rdf:about="#PersonalEvent"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:ID="hasTriggerType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasStringType">
  <rdfs:range rdf:resource="#String"/>
  <rdfs:domain rdf:resource="#StructureType"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:ID="hasStructureType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasModeling">
  <rdfs:domain rdf:resource="#ProcessPhase"/>
```

```
  <rdfs:range rdf:resource="#Modeling"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataTransformTask">
  <rdfs:range rdf:resource="#DataTransform"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasDataPreprocessing"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDateType">
  <rdfs:range rdf:resource="#Date"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasStructureType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#StructureType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasOutlierTask">
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
  <rdfs:range rdf:resource="#Outlier"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasDataPreprocessing"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasSetType">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasStructureType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#StructureType"/>
  <rdfs:range rdf:resource="#Set"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDomainObjectiveType"/>
<owl:ObjectProperty rdf:ID="hasSocialType">
  <rdfs:range rdf:resource="#Social"/>
  <rdfs:domain rdf:resource="#Environment"/>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:ID="hasEnvironmentType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasAlgorithm">
  <rdfs:subPropertyOf rdf:resource="#hasModeling"/>
  <rdfs:domain rdf:resource="#Modeling"/>
  <rdfs:range rdf:resource="#AlgorithmSelection"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasIntegerType">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasStructureType"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Integer"/>
  <rdfs:domain rdf:resource="#StructureType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasDataPreparation">
  <rdfs:domain rdf:resource="#Modeling"/>
  <rdfs:range rdf:resource="#DataPreparation"/>
  <rdfs:subPropertyOf rdf:resource="#hasModeling"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasNumberType">
  <rdfs:range rdf:resource="#Number"/>
  <rdfs:domain rdf:resource="#StructureType"/>
```

```
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasStructureType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataSource">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasData"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Source"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasInformationtype">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasData"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#Data"/>
  <rdfs:range rdf:resource="#InformationType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasTrainingSet">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#DataPreparation"/>
        <owl:Class rdf:about="#TrainingData"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:domain rdf:resource="#ModelWorkingData"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataPreparation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataSelectionTask">
  <rdfs:subPropertyOf rdf:resource="#hasDataUnderstand"/>
  <rdfs:domain rdf:resource="#DataSelection"/>
  <rdfs:range rdf:resource="#DataSelection"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataDescriptionTask">
  <rdfs:range rdf:resource="#DataDescription"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataUnderstand"/>
  <rdfs:domain rdf:resource="#DataUnderstand"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAttributeEvaluation">
  <rdfs:range rdf:resource="#AttributeEvaluate"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataPreparation"/>
  <rdfs:domain rdf:resource="#DataPreparation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasAttributeDerivationTask">
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasDataPreprocessing"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#AttributeDerivation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAlgorithmWorkingData">
  <rdfs:subPropertyOf rdf:resource="#hasAlgorithm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFinancialType">
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasEnvironmentType"/>
```

```
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Financial"/>
  <rdfs:domain rdf:resource="#Environment"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasExternalSource">
  <rdfs:range rdf:resource="#External"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataSource"/>
  <rdfs:domain rdf:resource="#Source"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasSocietyEvent">
  <rdfs:domain rdf:resource="#Trigger"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#InformationType"/>
        <owl:Class rdf:about="#Society"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasTriggerType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDataTranslationTask">
  <rdfs:range rdf:resource="#DataTranslation"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataUnderstand"/>
  <rdfs:domain rdf:resource="#DataTranslation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasOtherEvent">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#InformationType"/>
        <owl:Class rdf:about="#Other"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasTriggerType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#Trigger"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasConsumerEvent">
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasTriggerType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#Trigger"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#InformationType"/>
        <owl:Class rdf:about="#ConsumerEvent"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPsychographics">
  <rdfs:subPropertyOf>
```

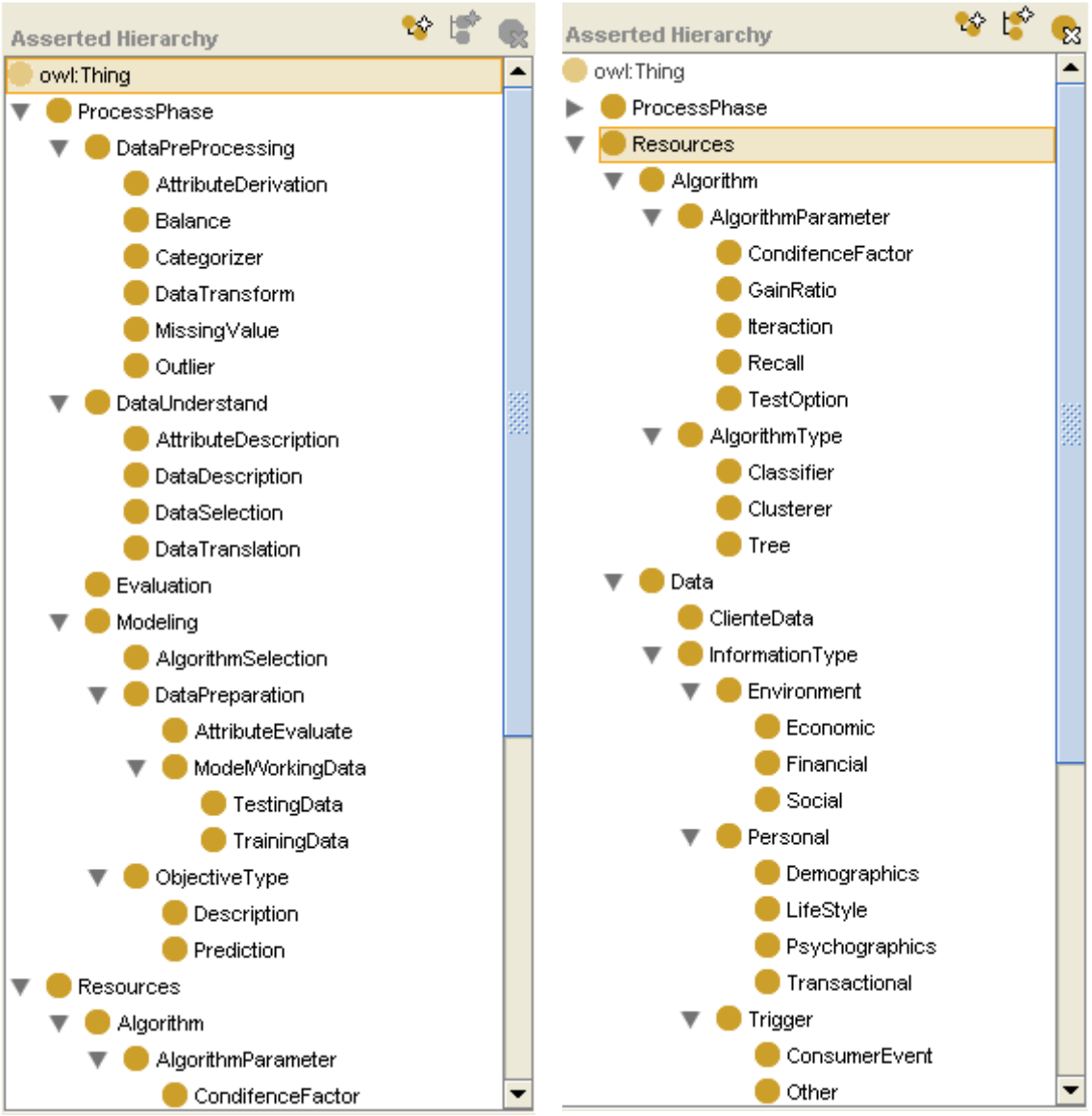
```
    <owl:TransitiveProperty rdf:ID="hasPersonalType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#Personal"/>
  <rdfs:range rdf:resource="#Psychographics"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEvaluationTecnique">
  <rdfs:subPropertyOf rdf:resource="#hasEvaluation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasTransactional">
  <rdfs:range rdf:resource="#Transactional"/>
  <rdfs:domain rdf:resource="#Personal"/>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasPersonalType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasInternalSource">
  <rdfs:domain rdf:resource="#Source"/>
  <rdfs:subPropertyOf rdf:resource="#hasDataSource"/>
  <rdfs:range rdf:resource="#Internal"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasCategorizationTask">
  <rdfs:subPropertyOf>
    <owl:ObjectProperty rdf:about="#hasDataPreprocessing"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
  <rdfs:range rdf:resource="#Categorizer"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasDataPreprocessing">
  <rdfs:range rdf:resource="#DataPreProcessing"/>
  <rdfs:domain rdf:resource="#ProcessPhase"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasData">
  <rdfs:domain rdf:resource="#Resources"/>
  <rdfs:range rdf:resource="#Data"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasDemographics">
  <rdfs:range rdf:resource="#Demographics"/>
  <rdfs:domain rdf:resource="#Personal"/>
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasPersonalType"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEconomicsType">
  <rdfs:subPropertyOf>
    <owl:TransitiveProperty rdf:about="#hasEnvironmentType"/>
  </rdfs:subPropertyOf>
  <rdfs:range rdf:resource="#Economic"/>
  <rdfs:domain rdf:resource="#Environment"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasMissingValueTask">
  <rdfs:subPropertyOf rdf:resource="#hasDataPreprocessing"/>
  <rdfs:range rdf:resource="#MissingValue"/>
  <rdfs:domain rdf:resource="#DataPreProcessing"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasStructureType">
  <rdfs:domain rdf:resource="#Data"/>
  <rdfs:range rdf:resource="#StructureType"/>
  <rdfs:subPropertyOf rdf:resource="#hasData"/>
```

```
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasAttributeDescriptionTask">
  <rdfs:subPropertyOf rdf:resource="#hasDataUnderstand"/>
  <rdfs:range rdf:resource="#AttributeDescription"/>
  <rdfs:domain rdf:resource="#DataUnderstand"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasLifeStyleData">
  <rdfs:range rdf:resource="#LifeStyle"/>
  <rdfs:subPropertyOf>
   <owl:TransitiveProperty rdf:about="#hasPersonalType"/>
  </rdfs:subPropertyOf>
  <rdfs:domain rdf:resource="#Personal"/>
 </owl:ObjectProperty>
 <owl:ObjectProperty rdf:ID="hasAlgorithmParameter">
  <rdfs:subPropertyOf rdf:resource="#hasAlgorithm"/>
 </owl:ObjectProperty>
 <owl:DatatypeProperty rdf:ID="hasRange"/>
 <owl:TransitiveProperty rdf:about="#hasPersonalType">
  <rdfs:range rdf:resource="#Personal"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#InformationType"/>
  <rdfs:subPropertyOf rdf:resource="#hasInformationtype"/>
 </owl:TransitiveProperty>
 <owl:TransitiveProperty rdf:about="#hasTriggerType">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:subPropertyOf rdf:resource="#hasInformationtype"/>
  <rdfs:range rdf:resource="#Trigger"/>
  <rdfs:domain rdf:resource="#InformationType"/>
 </owl:TransitiveProperty>
 <owl:TransitiveProperty rdf:about="#hasEnvironmentType">
  <rdfs:subPropertyOf rdf:resource="#hasInformationtype"/>
  <rdfs:domain rdf:resource="#InformationType"/>
  <rdfs:range rdf:resource="#Environment"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
 </owl:TransitiveProperty>
</rdf:RDF>


<!-- Created with Protege (with OWL Plugin 3.4, Build 533)  http://protege.stanford.edu -->
```

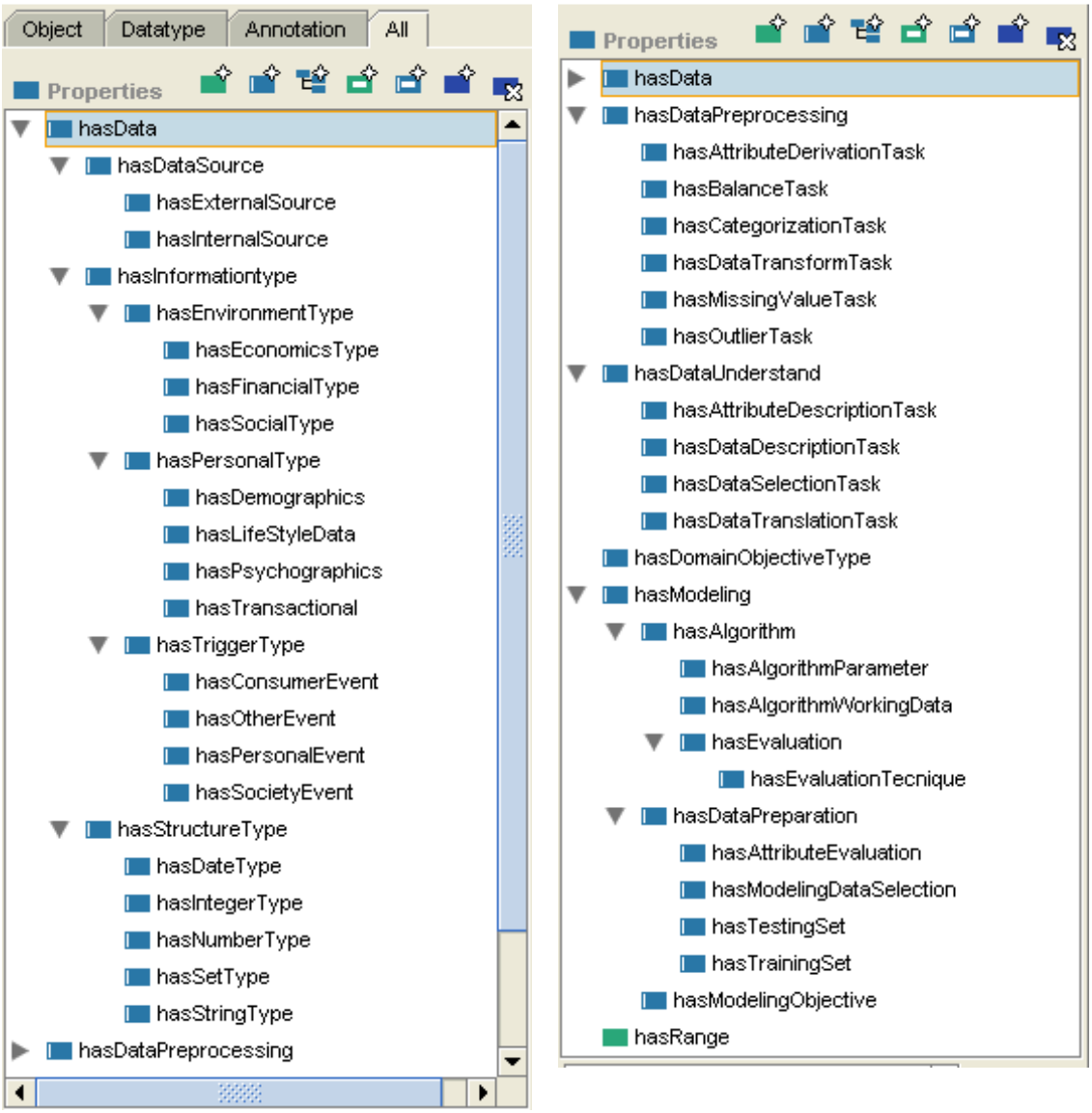
Appendix 4 - KDD ontology class hierarchy

print screen (partial view):

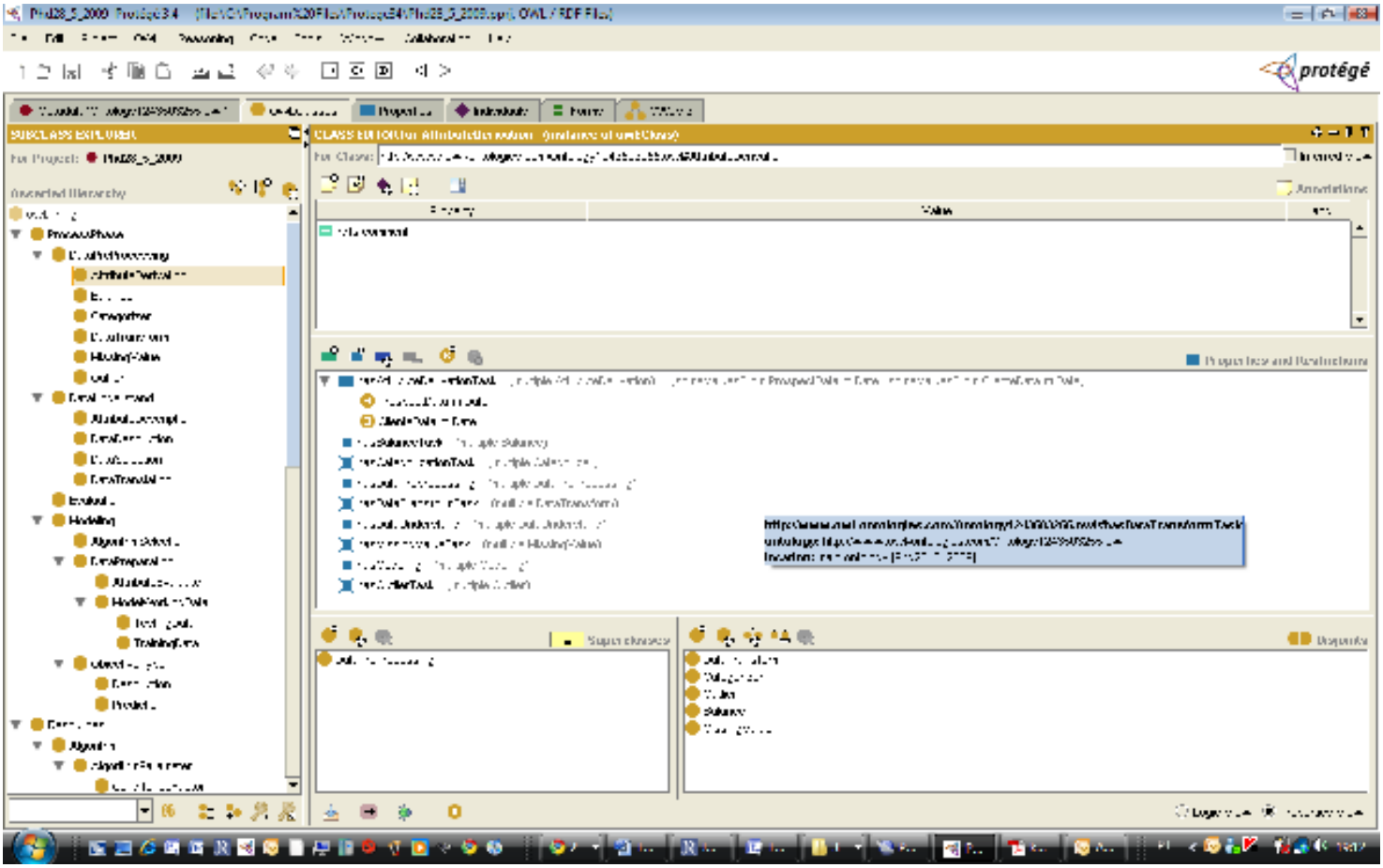


Attach 5 - KDD ontology Properties hierarchy

(partial view)



Appendix 6 – Protégé-Owl Tool Desktop



Appendix 7 – Expert Panel to Delphi Method

We have developed the Delphi method for database marketing related knowledge gathering. To do this, we have invited many people from a widespread of interests, such as, organizations (fuel oil distribution, insurance, cable TV, marketing agencies and banks), academics (professors, researchers and students) or practitioners (from small medium enterprises).

Therefore, our expert panel is composed by seven personalities, distributed as follows: 2 from oil company (from two different organizations); 1 from cable TV, 1 marketing agency professional; 1 marketing researcher; 1 marketing PhD student; 1 database marketing practitioner from a local small medium enterprise.

The communication process with the expert panel was made by e-mail. There wasn't any dialogue between the expert panel members.

This research inquiry was qualitative, through an open questions questionnaire and some other multiple choice questions. We have a double interaction method, in order to close each interaction, that is, after all answers received, we have made a synthesis and validated with expert panel.

First interaction: open questionnaire

How do you define database marketing?

Which are marketing activities that uses database marketing?

Closing First interaction: multiple choice

How do you define database marketing? (select two)

- A tool
- A marketing discipline
- A Customer Relationship Management technology
- A hip from computer scientists

Which are the marketing activities that uses database marketing? (select the five most important)

- Customer knowledge or identification
- Customer needs
- Customer wants
- Customer segmentation
- Customer categorization
- Customer profiling
- Cross selling
- Up-selling
- Cross marketing
- One-to-one marketing
- Customer reactivation
- Customer loyalty acquisition
- Customer fidelization
- Customer affiliation
- Segmentation
- Classification or clustering
- Sales prediction
- Customer upgrade
- Market basket analysis
- Prediction future behavior
- Description
- Churn
- Reactivation

Second interaction: Which is the main kind of data within marketing databases?

Which is the main kind of data within marketing databases?

Closing second interaction: multiple choice

Since most of experts have responded through database attribute examples, we have performed a common selection and "typified" the attributes into marketing data types and presented them to expert panel:

Marketing database type information: select the most important (some examples of each one are presented):

Psychographics: personal data that can easily be changed.

- monthly income
- professional occupation
- scholarship

Demographics: physical and personal data that is almost definitive and almost never changes.

- gender
- marital status
- birth date
- children
- race

Transactional: consumer based information regarding its commercial activity

- monthly consumption
- number transactions/month
- number items/month
- shops visited
- promotional acceptance

Lifestyle or behavior: consumer or social related information.

- hobbies
- car type
- holidays
- club membership

In addition to the above customer oriented data types there are two other groups of data:

Market data: environmental market data

- Financial (e.g., inflation tax rate)
- Market (e.g., market or product share)
- Social (e.g., national birth, death or other sensus)

Trigger events data:

- Consumer (e.g., marital status change or children number)

- life related (e.g., new car or new house)
- others (e.g., accident, prison, tax penalties)

Third interaction: open question?

Which are main database marketing operations?

Closing third interaction: multiple choice

Since most experts have responded through operational examples, without denoting any sequential order, we have performed a selection and proposed and organized a form for the database marketing process

Since database marketing has a set of operations (mostly unidentified by the panel group) do you agree with this process order?

- Marketing objectives definition
- Data collection and selection
- Data preparation
- Data pre-processing
- Modeling
- Deployment
- Evaluation