Considering Application Domain Ontologies for Data Mining

FILIPE MOTA PINTO
Computer Science Department of School of technology and Management
Polytechnic Institute of Leiria
Morro do Lena – Leiria
PORTUGAL
fpinto@estg.ipleiria.pt

MANUEL FILIPE SANTOS
Department of Information Systems of Engineering School
University of Minho
Campus de Azurém – Guimarães
PORTUGAL
mfs@dsi.uminho.pt

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.

Key-Words: - Ontologies, Knowledge Discovery, Databases, Data Mining.

1 Introduction

In artificial intelligence, ontology is defined as a specification of a conceptualization [17]. Ontology specifies at a higher level, the classes of concepts that are relevant to the domain and the relations that exist between these classes [24] [28]. 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 [30].

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 [15].

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 [3] [24].

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 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.

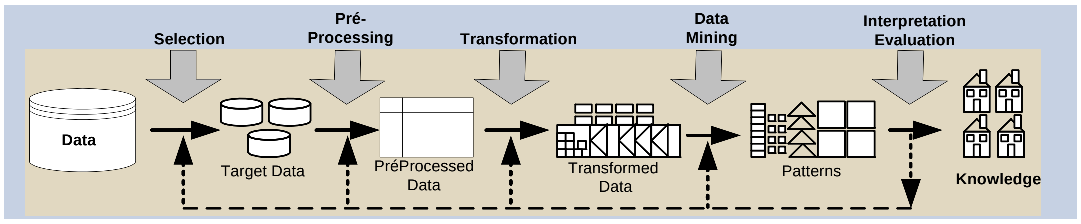

Figure 1: Knowledge discovery process framework adapted from [10]

KDD process is usually performed by experts who use their own knowledge for selecting the most relevant data in order to achieve domain objectives [16]. 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.

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 Predictive Model Markup Language

Predictive model markup language (PMML) is an XML-based language that provides a way for applications to define statistical and data mining models and to share these models between PMML compliant applications (Data Mining Group). Furthermore, the language can describe some of the operations required for cleaning and transforming input data prior to modeling. Since PMML is an XML based standard, its specification comes in the form of an XML schema that defines language primitives as follows [6]:

- Data Dictionary;
- Mining schema;
- Transformations.
- Model statistics;
- Data mining model.

2.3 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[1]).
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) [18]. SWRL is a rule language that combines OWL with the rule markup language providing a rule language compatible with OWL.

2.4. Semantic Web Language Rule

At the best of our knowledge there are no standard OWL-based query languages. Several RDF[2]-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 [5] [19]. 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 [24]. 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:

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

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 [2] [3] [9] 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[3] or AXIS[4]. 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 [14]. 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 the Data Mining 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 work we focus the role of ontology in order to assist the KDD in different stages of the process: 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

[1]http:// www.w3c.org
[2] Resource Description Framework

[3] http://olp.dfki.de/pkdd04/cfp.htm
[4] http://ralyx.inria.fr/2006/Raweb/axis/uid4.html

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 [12] [4]. 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.

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

Following METHONTOLOGY we had constructed our ontology in terms of process assistance role. Nevertheless, domain concepts and relations between concepts were introduced according some literature directives [4][31].

In order to formalize our domain concepts we have used some relevant scientific KDD [27] [10, 11] [1] and ontologies [23] [21] literature. However, whenever some vocabulary is missing it is possible to develop a research method (e.g., through Delphi method [8] [7] [25, 26]) in order to achieve such domain knowledge thesaurus.

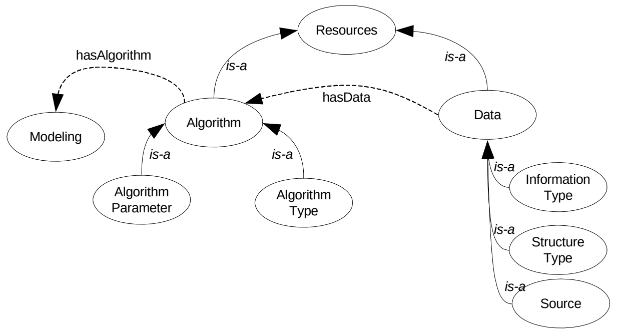

Figure 3 – ontology class hierarchy and relationships (partial view)

At the end of this methodology first step we have identified the following main classes:

- Resource
  - Data
    - Attribute
      - Information Type
      - Structure;
    - Source
  - Algorithm;
    - Type;
    - Parameters;
- ProcessPhase
  - Data Pre-Process
  - Data Understand
  - Modeling;
    - Objective Type
    - Data Select
  - Evaluation
- ResultModel

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):

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

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

iii. *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 already tested concepts from other data mining ontology (DMO) [21], 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.

The DMO uses a service named Abstract Data Mining Service that simplifies its architecture as the realization of the OWL service with a detailed description of its profile and model. DMO has three essential types of data mining components involved in the assisted KDD process: DM-element, DM-task and DM-service, which in our case correspond to Resource, Process and Phase. Such DM-elements are represented by OWL classes together with variations of their representations in XML (allowing information interchange with PMML DM models). It means that a concept described by an OWL class can have one or more related XML schemas that define its concrete representation in XML.

In the DMO, for simplicity reasons, there are two defined types of DM-elements: settings and results, which in our case correspond to *Algorithm* and *Data* classes. The settings represent inputs for the DM-tasks, and on the other hand, the results represent outputs produced by these tasks. There is no difference between inputs and outputs because it is obvious that an output from one process can be used, at the same time, as an input for another process.

The 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 [32] 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.

iv. *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/DBMiPhDf
pinto.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="InformationType"/>
    </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 hasAlgorithm 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 PersonalInformationType(Demographics)
→ *hasCheckConsistency*

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

```
? Attribute hasSource
      attribute hasSource (CustomerTable).
? Attribute hasInformation Type:
Attribute hasInformationType (Personal) then:
      attribute hasPersonalInformationType(Demographics)
? Attribute hasStructureType
      attribute hasStructureType (Date).
: attribute hasStructureType(Date) AND
            PersonalInformationType(Demographics) then:
 : attribute (Demographics; Date) hasDataPreparation
 : attribute (Demographics; Date) hasDataPre-Processing
                AND Check missing values
                AND Check outliers
                AND Check consistency
                AND deriveNewAttribute
```

In this 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 learning cicle

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.

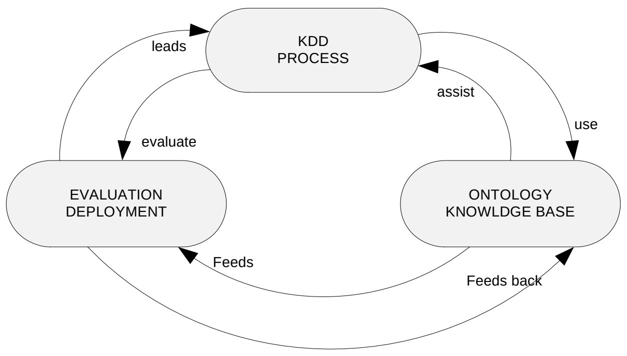

Figure 5 – ontology learning cycle

As depicted in figure 5, 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.

5. System Prototype

A general overview of the main components of the system is shown in Figure 4. Our system has four main components:

- ***Knowledge base***: developed over Protégé[5] 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 [21];
- ***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 [22]. We need implement engine or map to the existing rule engine, here the bridge;

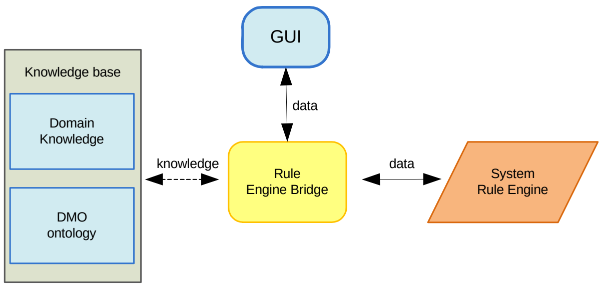

Figure 4: DBMI system components

- ***System rule engine:*** is based on SWRL API supported by Jena Toolkit [20] 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 [13];
- ***GUI***: developed through Eclipse java software to develop it supports the system user interface.

Keeping it straightforward, the assistant system communicates over the rule engine bridge with the Pellet reasoner, which is able to answer a subset of SWRL/SPARQL queries [29]. 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.

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

[5] http://Protege.stanford.edu

simplicity reasons, trough Eclipse software6 it is implemented an application able to navigate a user in the KDD process.

.

[6] www.eclipse.org

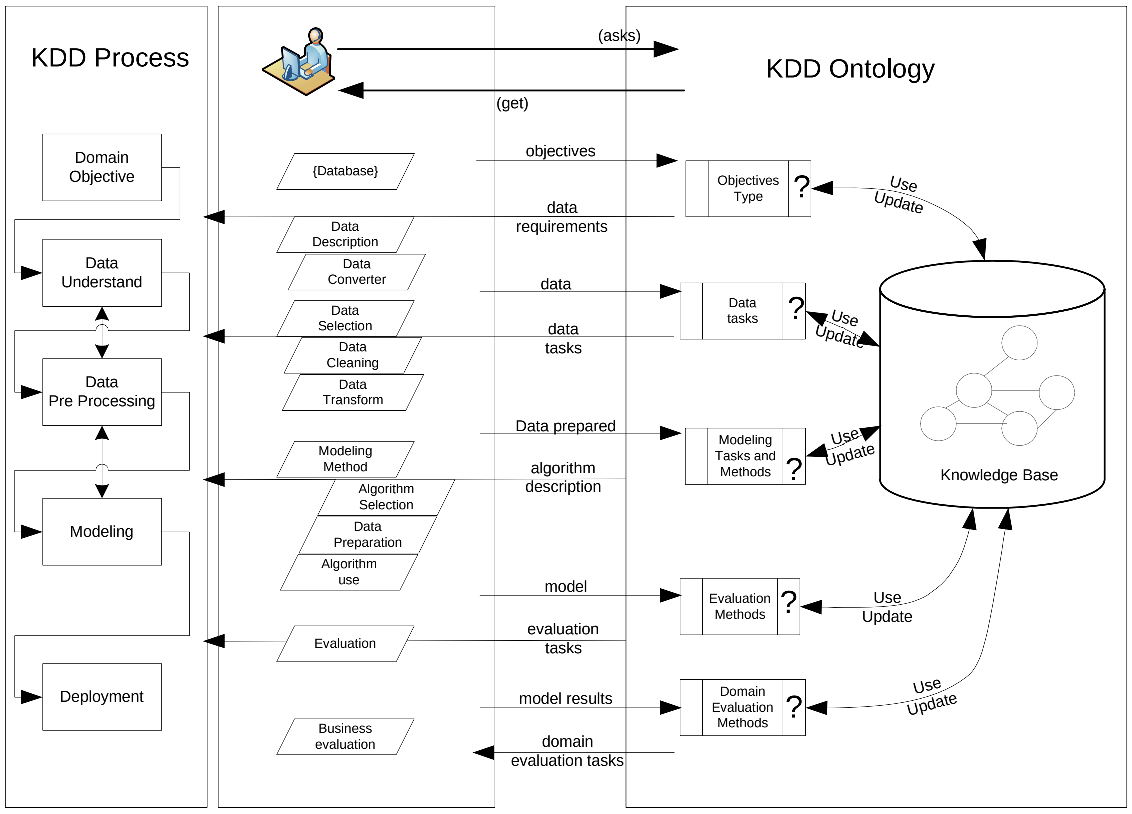

Figure 5 – Ontology assistance to knowledge discovery process

Our main objective is 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 layers (Figure 5): KDD running process phases; user interface layer; 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., *firstTransactioDate*, *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*"[10]. As example of such capability, the following expression, demonstrates how each phase may be connect through an inference instruction:

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

This SWRL *get* expression gets the set of modeling tasks to be performed 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*). Such set of specifications will invoke the knowledge base through the inference process.

hasDataType, is a OWL property that links the objective type and data types previously used and registered at knowledge base. *DataSet* is the expression output with workable set of data. Each variable is preceded by a question mark.

6. Experiment

Our system prototype operation follows general KDD framework (Figure 1) and uses the ontology to assist at each user interaction.

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; status; gender; vehicleType; vehicleYear; fuelType }
TransactionTable {
idMov; idCard, date, fuelValue, fuelLitres; shopValue; shoppUnits; stationcode}
StationTable{

```
        stationCode; stationType; postCode}
```

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

6.1. 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)
```

where do became:

```
                 do="classification"
```

6.2. 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:

- hasMissingValue(?att) : performs a data completeness evaluation in terms of e.g., missing values;
- 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;

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

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

This attribute will be assigned a record as:

```
{
   : Information type : #Personal
   : Personal informationType: #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")
```

6.3. Data Preprocessing

Since we have selected working data (*wds*) we need 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*.

6.2.4 Modeling

At modeling phase the objective is to modelate the data using previously selected and prepared data set throughout algorithms modeling.

Such work may vary from single algorithm use (direct use of e.g., k-means cluster algorithm) or some complex algorithms use (e.g., self organizing maps neural networks in conjunction with C5.0 decision tree).

At modeling phase there exists several interactions looking for the best algorithm combination (parameters settings definition and attribute selection) towards the best model performance.

Modeling phase instantiation is the ending section of a set of user decisions:

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

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

Regarding our running example, we have

```
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 }
hasTraningSet= 66,6% (6 183 records)
hasTestingSet= 33,3% (3 102 records)
hasAlgoritm(alg)→C5.0
workingDataSet(wds) → hasModel(?m)
```

then we have found to *?m*:

```
        if (age<27 and
          vehicleType="Lig" and
          sex="Female") then
        1stUsed<10
```

At this model we may say that, female card owners with less than 27 years old and has a car "ligeiro" category, that it would use fuel station located in range than 10 kilometers than her address.

6.5. Deployment

Each running KDD process must be evaluated according to the results, in order to be updated and reused in latter projects.

Firstly we need to perform a model evaluation, through an evaluation method. Such evaluation will depend of which type of model we have, which algorithm was used and of course, which model do we have. Then evaluate it through evaluation algorithms available, e.g., AUC (area under curve) or PCC (principal components analysis):

```
: getEvaluation[Model?m)^
   hasModeling(?met)^
   hasAlgorithm(?alg)^
   hasEvaluation(?m,?met,?alg)
                -> Evaluation (?m,?ev)]
```

To answer the question: "customer profile according fuel station use" we have got the model which now must be evaluated. We use four-fold table to evaluate obtained model:

Test Set dimension: 3102

		Customer Profile	
		TP	FP
Station	TP	2 435	85
	FP	409	173

Accuracy:	84,07%
Sensibility:	29,72%

Therefore this model will be added to the ontology as:

```
Model(c5.0)^
modelingObjectiveType(classifiation)^
hasWorkingData(idCard; idCliente; birthDate; age;
    initialCardDate; cardAge…;
   1stUsed, 2stUsed;3stUsed ) ^
Evaluation(PCC; 0,8407; 0,2972)->
  hasResultMoldel (c5;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.

7 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:

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In International Conference on Management of Data, editor, *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216. International Conference on Management of Data, 1993.

[2] Sarabjot Singh Anand, Marko Grobelnik, Frank Herrmann, Mark Hornick, Christoph Lingenfelder, Niall Rooney, and Dietrich Wettschereck. Knowledge discovery standards. *Artificial Intelligence Review*, 27(1):21–56, 2007.

[3] Abraham Bernstein, Foster Provost, and Shawndra Hill. 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), 2005.

[4] M. Blazquez, M. Fernandez, J. M. Garcia-Pinar, and A. Gomez-Perez. Building ontologies at the knowledge level using the ontology design environment. In *Knowledge Acquisition Workshops and Archives*, Voyager Inn, Banff, Alberta, Canada, 1998. University of Calgary.

[5] Ana Maria Borges, Marla Corniel, Richard Gil, Leonardo Contreras, and Rafael H. Borges. Towards a study opportunities recommender system in ontological principles-based on semantic web environment. *WSEAS Transactions on Computers*, 8(2):279–291, 2009.

[6] Peter Brezany, Ivan Janciak, and A Min Tjoa. *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, 2008.

[7] Hui-Chun Chu and Gwo-Jen Hwang. A delphi-based approach to developing expert systems with the cooperation of multiple experts. *Expert Systems with Applications*, 34:2826–2840, 2008.

[8] Andre L. Delbecq, Andrew H. Van De Ven, and David H. Gustafson. *Group Techniques for Planning- A Guide to Nominal Group and Delphi Processes*. Scott, 1975.

[9] Pedro Domingos. Prospects and challenges for multi-relational data mining. *SIGKDD Explorer Newsletter*, 5(1):80–83, 2003.

[10] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. In AI Magazine, 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, 1996. American Association for Artificial Intelligence.

[11] Usama Fayyad and Ramasamy Uthurusamy. Data mining and knowledge discovery in databases. In ACM, editor, *Communications of the ACM*, volume 39, pages 24–26, New York, NY, USA, 1996. ACM.

[12] Mariano Fernandez, Asuncion Gomez-Perez, and Natalia Juristo. Methontology: From ontological art towards ontological engineering. Technical report, AAAI, 1997.

[13] Ernest Friedman-Hill and David Scuse. Jess: The rule engine for the java platform. Technical report, Sandia National Laboratories, 2008.

[14] Asuncion Gomez-Perez, Mariano Fernandez-Lopez, and Oscar Corcho. *Ontological engineering*. Springer, 2nd edition, 2004.

[15] Paulo Gottgtroy, Nik Kasabov, and Stephen MacDonell. An ontology driven approach for knowledge discovery in biomedicine. 2004.

[16] Paulo Gottgtroy, Nik Kasabov, and Stephen Macdonelll. 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., 2003.

[17] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[18] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language - combining owl and ruleml. Technical report, W3C, 2004.

[19] Daniel Hunyadi and Iulian Pah. Ontology used in a e-learning multi-agent architecture. *WSEAS Transactions Information Sciense and Application*, 5(8):1302–1312, 2008.

[20] Brian McBride. JENA: A semantic web toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.

[21] Hector Oscar Nigro, Sandra Gonzalez Cisaro, and Daniel Xodo. *Data Mining with Ontologies: Implementations, Findings and Frameworks*. Information Science Reference. Information Science Reference - IGI Global, London, igi global edition, 2008.

[22] Bijan Parsia and Evren Sirin. Pellet: An owl dl reasoner. In *3rd International Semantic Web Conference (ISWC2004)*, 2004.

[23] Joseph Phillips and Bruce G. Buchanan. 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, Oct 2001.

[24] 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.

[25] 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.

[26] Filipe Mota Pinto, Alzira Marques, and Manuel Filipe Santos. Database marketing process supported by ontologies: System architecture proposal. In *11th International Conference on Enterprise Information Systems*, 2009.

[27] Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[28] Alberto Salguero, Francisco Araque, and Cecilia Delgado. Ontology based framework for data integration. *WSEAS Transactions Information Sciense and Application*, 5(6):953–962, 2008.

[29] Andy Seaborne. RDQL - a query language for rdf. Technical report, W3C, 2004.

[30] Chia-Cheng Shen and Huan-Ming Chuang. A study on the applications of data mining techniques to enhance customer lifetime value. *WSEAS Transactions Information Science and Applications*, 6(2):319–328, 2009.

[31] Reid G. Smith and Adam Farquhar. The road ahead for knowledge management: An AI perspective. *American Association for Artificial Intelligence*, 1:17–40, 2008.

[32] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Technique*. The Morgan Kaufmann Series in Data Management Systems, 2nd edition, 2000.