ANALYSIS OF ONTOLOGICAL INSTANCES
A Data Warehouse for the Semantic Web

Roxana Danger and Rafael Berlanga
Department of Informatics Languages and Systems, Universitat Jaume I

Keywords: Semantic Web, Data Warehouse, Description Logic.

Abstract: New data warehouse tools for Semantic Web are becoming more and more necessary. The present paper formalizes one such a tool considering, on the one hand, the semantics and theorical foundations of Description Logic and, on the other hand, the current developments of information data generalization. The presented model is constituted by dimensions and multidimensional schemata and spaces. An algorithm to retrieve interesting spaces according to the data distribution is also proposed. Some ideas from Data Mining techniques are incorporated in order to allow users to discover knowledge from the Semantic Web.

1 TOWARD PATTERN RECOGNITION IN THE SEMANTIC WEB

The *Semantic Web* is a new form of web conceived for allowing human users and software tools to process and share the same sources of information. The Semantic Web relies on a set of standards which provide syntactic consistency and semantic value to all of its content. For example, Description Logic is used as the theoretic base for the description of web items, and the languages RDF and OWL for their syntactic representation. Description Logic defines a family of knowledge representation languages which can be used to represent, in a well-understood formal way, the knowledge of an application domain. This knowledge, known as ontology, ranges over the terminological cognition of the domain (the interesting object classes, or concepts, its $Tbox$) and its examples (the instances of the object classes, its *Abox*).

Data analysis in the Semantic Web will be the most important process when the population of ontologies[1] becomes a reality. Its final goal is the recognition of patterns amongst the values of the attributes of the ontological instances, which could turn into knowledge or allow its discovery. OWL tools allow users to create new concepts related to one or more existing ontologies, and to determine the instances associated to such concepts.

An additional tool is needed if we want to perform in a versatile way customized data analysis, either full or partial, so that each object can be studied from different points of view focusing on distinct particular features. Such a tool should allow users to navigate through an instance set and its properties, being able to discriminate between relevant and superfluous information. Moreover, it should compute and display statistical indexes able to describe and report about the extracted patterns.

The formalization of such a tool, following the framework described below, is the purpose of the present work. Starting from an available ontology, this is enriched with information provided by the data analyst, for example specifying the *atomic data combination functions*. These functions provide the way for combining atomic data to form a generalized instance representing a set of instances. Then, two main structures have to be built: the conceptual dimensions and the multidimensional conceptual spaces. The conceptual dimensions are partial order specifications between objects, which allow to browse through their semantic relations. The multidimensional conceptual

[1] The population of ontologies is the process of adding instances to an ontology in order to enrich it with examples of its domain knowledge.

Danger R. and Berlanga R. (2007).
ANALYSIS OF ONTOLOGICAL INSTANCES - A Data Warehouse for the Semantic Web.
In *Proceedings of the Second International Conference on Software and Data Technologies - Volume ISDM/WsEHST/DC*, pages 13-20
DOI: 10.5220/0001332200130020

spaces can be seen as “intelligent object containers”. They make use of a subset of conceptual dimensions, a specification of relevant abstraction levels and a set of atomic data combination functions in order to (re)construct appropriate generalized instances. Different statistical indexes (e.g. frequencies), associated to the conceptual dimensions, can be used to characterize patterns in the conceptual spaces. The most suitable data analysis technique for carrying out this proposal is *data warehousing*.

The present work is not the first attempt to formalize a data warehouse for the Semantic Web. Within the Data Warehouse Quality (DWQ) project (Hacid and Sattler, 1998) a formalization for the multidimensional modeling based on an extension of the constructors of description logic is proposed. In this way, new object classes could be described by specifying aggregability operations, and the traditional reasoning over ontological instances could be applied. However, the demonstration of the undecidability of minimal languages that operate with aggregate operators(Baader and Sattler, 2003) makes the proposal of the DWQ project unfeasible.

On the other hand, the ideas of the traditional data warehouse (and OLAP techniques) has been extended to object oriented modeling, (Buzydlowski et al., 1998; Trujillo et al., 2001; Nguyen et al., 2000; Binh and Tjoa, 2001; Abelló, 2002). Considering that description logic was designed as an extension to frames and semantic networks, the basis of object-oriented data warehouse could be applied in order to define a data warehouse for the Semantic Web. However, the flexibility of object-oriented formalization causes a more sparse structure in object-oriented databases that in traditional ones. Moreover, the restrictions of OLAP implementations drastically reduce the useful set of objects to be used in the analysis.

Unlike these previous works, this paper proposes a multidimensional model for the analysis of ontological instances that merge both approaches. The idea is the creation of meta-ontologies in order to enrich the knowledge of ontologies with data analysis information. This data analysis information focuses on the description of interesting object classes and on the aggregation process. The reasoning of description logic is used in a preliminary phase to 1) recover the satisfiable[2] object classes that can be used on analysis processes, 2) discover the hierarchical and aggregate orders between the classes, and 3) assign each instance to the set of object classes to which it belongs.

This paper describe our proposal in detail. Firstly, the data analysis information is introduced. Then, the proposed model is described, starting from the definition of dimensions and their operators (section 3) and following with the specification of the multidimensional conceptual space (section 4). The two following sections are focused on the extraction of interesting conceptual spaces and their use, respectively. The last section gives some conclusions and future work.

2 ANALYSIS METADATA

Information descriptions useful for the analysis are those available in the ontologies in form of instances. However, they are not enough to analyze data and discover patterns. New interesting concepts and particular issues related to the generalization process are essential in order to generate descriptions that represent relevant and realistic visions of the application domains of the analyzed ontologies. We call all this information *analysis metadata*, which comprises the following elements:

- *description of new concepts*, which it is used to introduce additional levels of abstraction in the concept hierarchies expressed in an ontology, and/or to link concepts from different ontologies. New concepts may be obtained extending old ones with paths to previously unrelated concepts. They can also semantically represent hierarchical clusters obtained using clustering algorithms.
- *description of the combination functions* (see definition below); it is used to specify ways for generalizing sets of data of the same type during the instance generalization process. The data analyst is responsible for deciding the combination functions that are semantically suitable for a given data set. For example, the combination function which computes the average of a set of values is semantically suitable for a temporal sequence of temperatures of a town, but not for a set of temperatures of different towns.

Although it is perfectly plausible to define such descriptions for every new multidimensional conceptual space, a better solution is to keep this semantic information always available and to apply it according to the requirements of each case. This goal can be achieved building a meta-ontology containing the sort of information described above, again using Description Logic. In this way, analysts can proceed more efficiently as they can reuse the analysis metadata. Even more importantly, in this way the coherence of different studies is granted, providing an ontology with an intrinsic robustness toward analysis processes. Thus,

[2] A concept (or object class) is satisfiable if it is consistent and there exists an interpretation on which appears at least an instance of this concept.

further studies can be more easily performed by comparing different analysis on the same knowledge domain and/or the point of views of different analysts.

The description of the combination functions can be specified through instances associated to the notion of *Combinable Concept* of this meta-ontology:

$$\begin{aligned}&CombinableConcept \equiv\\ &\quad \equiv \exists hasConcept.URI \sqcap \forall hasRelation.URI \sqcap\\ &\quad \exists hasCombinationFunction.CombinationFunction\\ &CombinationFunction \sqsubseteq\\ &\quad \sqsubseteq \exists hasName.String \sqcap \exists hasImplementation.URI\end{aligned}$$

Combinable concepts are those for which a combination function can be defined. A combinable concept can be a datatype, a named concept (defined via a URI), or a concept derived from a composition of relations beginning with a named concept (specified by the URI where the start concept is defined and the relations from it).

3 DIMENSIONS AND THEIR OPERATIONS

A dimension is described by a set of concepts and the way to browse through them. Such browsing is performed using the operators of abstraction and generalization between ontological instances, and the selection operators defined below.

We consider that an abstract ontology is constituted only by the terminological knowledge. An ontology that contains a set of instance axioms (the *Abox of the ontology*, composed by axioms specifying the class C of an instance a -$C(a)$- or the relations between two instances a and b -$R(a,b)$-) is called concrete ontology. As it is usual in description logic, the interpretation of the ontology is $(I = \Delta^I, .^I)$, where Δ^I denotes the set of instances belonging to an ontology $\mathcal{O}$, and $.^I$ the interpretation of the concepts defined on $\mathcal{O}$; $I \Vdash x$ represents that x is deduced from I; $\top$ represents the top concept: *thing*. We denote with $\mathcal{A}$ the Abox of $\mathcal{O}$, with $\mathcal{R}$ the set of axioms associated with the relations of an ontology, with N_C and N_R the set of named concepts and relations of the ontology, respectively. Besides, the interpretation of a datatype is defined by $I^D = (\Phi^D, .^D)$, where Φ^D denotes the data set belonging to a datatype, Φ, of the ontology, and $.^D$ associates each datatype Φ with a strict subset of data in Φ^D, $\Delta^I \cap \Phi^D = \emptyset$. All representable data in the ontology belongs to the set $\mathcal{U} = \cdot^I \cup \bigoplus^{\mathcal{D}}$.

The definition of path between two concepts C and C' and that of dimensional partial order are given below. Intuitively, the former is the set of lists of relation-concept pairs that links C and C' by using consistent ontological definitions, and the latter is used to relate concepts using both the aggregate (as defined bellow) and the hierarchical order between concepts implicitly defined in a given ontology $\mathcal{O}$.

Definition 1. $Path(C,C^{'}) = \oplus_{1 \leq i \leq n} \langle RS_i, C_i \rangle$ is an aggregation path from concept C to concept $C^{'}$ of the ontology $\mathcal{O}$ if $C_1 = C, C_n = C^{'}$, and there exists an interpretation $I = (\Delta^I, .^I)$ of $\mathcal{O}$ such that $\exists x_i \in \Delta^I, 0 \leq i \leq n$, such that $x_0 \in C^I$ y $x_i \in C_i^I, \langle x_{i-1}, x_i \rangle \in R_i^I$, for $1 \leq i \leq n, R_i \in RS_i$.

The process of path retrieval must be exhaustive enough to allow the recovering of all aggregation relations between two concepts. It can be performed by using an extension of the tableau algorithm for the SHOIQ(D) language (Danger, 2007). It is worth emphasizing that these paths not only describe the aggregation relations between two concepts, but also the aggregation order between all concepts of an ontology.

Definition 2. Let $\mathcal{O}$ be an ontology. A dimensional partial order, denoted as $\xrightarrow{\sqsubseteq}$ is a partial order between all possible pairs of concepts $C, C' \in N_C$ defined according to the following constraints:

- $C \xrightarrow{\sqsubseteq} C'$ if $C' \sqsubset C$, or
- $C \xrightarrow{\sqsubseteq} C'$ if $\exists Path(C,C')$

The symbol $\xrightarrow{\sqsubseteq}*$ is the reflexive and transitive closure for relation $\xrightarrow{\sqsubseteq}$.

Definition 3. Let $\mathcal{O}$ be an ontology. The pair $D = (C_d, \xrightarrow{\sqsubseteq})$ is a conceptual dimension, being C_d a set of satisfiable concepts in $\mathcal{O}$, $\top \in C_d$ and $\xrightarrow{\sqsubseteq}$ the relation of dimensional partial order for the elements in C_d.

Example 1. In Figure 1 a *workplace* dimension which combines hierarchical and aggregation relations is shown. This dimension can be used to identify a specific place with different levels of granularity.

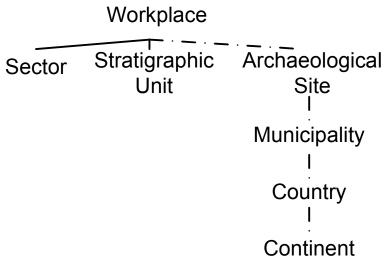

Figure 1: *Workplace.* dimension.

Operations

The ontological instances of each dimension can be represented by using different point of views of (concepts associated with) the dimension. It is thus necessary to define two different kinds of operations over such instances. The first one is the selection operator, used to specify the interest portion of the instance that must be shown (for example, when the concept represented by a dimension is replaced by a concept related to the first one by an aggregate relation). The second important operation is the generalization, used to generalize a set of instances (for example, when the concept represented by a dimension is replaced by a concept related with the first one by a hierarchical relation). The following definitions formalize these operators.

Definition 4. Let $\mathcal{O}$ be a concrete ontology with Abox $\mathcal{A}$; the description of an instance $a \in \mathcal{A}$ is the set $d(a) = \{R(a,b), R(a,b) \in \mathcal{A}\}$. This instance is said to be of type C if C is the most specific concept that can be deducted from I for a, i.e., $\forall C*$ such that $I \Vdash C*(a), C \sqsubset C*$.

Definition 5. Instance a' is called the specialization of an instance a of class C towards class C', if its description $d(a \uparrow_{C'} a')$ is not undefined, and if $\mathcal{A} \setminus C(a) \cup \{C'(a'), d(a)\} \cup d(a \uparrow_{C'} a')$ is consistent. The description of $d(a \uparrow_{C'} a')$, is defined as follows:

$$\begin{array}{ll} \begin{array}{l} \{R'(a',b') | R(a,b) \in d(a), \\ \quad fe(C,C')(R) = \{\langle R',C'\rangle\}, \\ \quad\quad d(b \uparrow_{C''} b') \neq undefined\}, \end{array} & \begin{array}{l} \text{if } \exists f_e(C,C'), \\ \quad |d(a \uparrow_{C'} a')| = |d(a)| \end{array} \\ \\ undefined, & \text{otherwise} \end{array}$$

where f_e is a specialization function of the concept C to the concept C'. This function defines how to transform each relation on the abstract concept to the appropriate relation on the specialized concept (Danger, 2007).

The operation of abstraction of an instance, denoted by $d(a \downarrow_{C'} a')$, can be defined in a similar way.

Definition 6. Let $\mathcal{O}$ be an ontology. Let ℓ be an undefined data that represent any data in $\mathcal{U}$. A pseudo-instance a[3] of type C is a selector if its description, $d(a)$, satisfies that:

$$\begin{array}{l} \forall b | \exists \{R_1,\ldots,R_n\} \subseteq N_R, R_1(a,a_1),\ldots,R_{n-1}(a_{n-1},a_n), \\ \quad R_n(a_n,b) \in d(a) \Rightarrow \\ \quad b \in \{\ell\} \bigcup\limits_{\substack{\forall C', \exists Path(C,C') \text{and } R_1,\ldots,R_n \\ \text{is the order of the relations on } Path(C,C')}} C'^I. \end{array}$$

Definition 7. Let $\mathcal{O}$ be a concrete ontology with Abox $\mathcal{A}$, $a \in \mathcal{U}$. Let ℓ be an undefined data that represent any data in $\mathcal{U}$. An instance $a' \in \mathcal{U}$ is selected by an instance selector a if:

- $a' \in \Phi$, $a' \in \{a, \ell\}$ or
- $a \in C^I, C \in N_C$ y a'' computed for $d(a' \downarrow_{C^I} a'')$ is such that:
  $$\begin{array}{l} \forall b \in \Phi \cup \{\ell\} \text{ such that } \exists R_1(a,a_1),\ldots, \\ \quad R_{n-1}(a_{n-1},a_n), R_n(a_n,b) \in d(a) \Rightarrow \\ \exists R_1(a'',a''_1),\ldots,R_{n-1}(a''_{n-1},a''_n), R_n(a''_n,b'') \in \\ \quad \in d(a'') \wedge (b'' = b \vee b = \ell) \end{array}$$

Example 2. In Figure 2 a fragment of an archeology ontology is represented. A selector instance a constituted by the set {*has_morphology(a, a'), has_group(a', "open"), has_order(a', ℓ), has_decoration(a, ℓ), has_color(a, "gray")* } allows the users to recover from the ontology the descriptive fragments of *ceramic artifacts* instances according to the properties *group*, *order*, *decoration* and *color*, but notice that the morphologic group of the ceramic must be open, and its color gray.

Definition 8. Let c_{Φ^D} be a function (called a combination of simple data) which allows each datatype Φ to be mapped to another function rep_Φ, which in turn maps subsets of Φ in *a compact representation of the input subset*[4]. Let d, d' be two data in $\Delta^I \cup \Phi^D$. A complete combination of data d and d' is the data $d \dot{\cup}_t d'$ computed as follows:

$$\begin{array}{ll} c_{\Phi^D}(\Phi)(\{d,d'\}), & \text{if } \{d,d'\} \subseteq \Phi, \\ \\ d' \dot{\cup}_t d, & \begin{array}{l} \text{if } d \in_* C, d' \in_* C', \\ \quad C \sqsubseteq C' \end{array} \\ \\ \begin{array}{l} d(d \uparrow_{C'} d') \cup d(d') \setminus \\ \quad \setminus \{R(d',b_1),\ldots,R(d',b_n) | \\ \quad\quad \{b_1,\ldots,b_n\} \subseteq_* C''\} \cup \\ \quad\quad \cup \{R(d', b_1 \dot{\cup}_t \ldots \dot{\cup}_t b_n) | \\ \quad\quad\quad \{R(d',b_1),\ldots,R(d',b_n) | \\ \quad\quad\quad\quad \{b_1,\ldots,b_n\} \subseteq_* C''\} \end{array} & \begin{array}{l} \text{if } d \in_* C, d' \in_* C', \\ \quad C' \sqsubseteq C \text{ and} \\ \quad \text{during the process} \\ \quad \text{no indefinitions} \\ \quad \text{are obtained} \end{array} \\ \\ undefined, & \text{otherwise} \end{array}$$

In particular, two types of functions for data combination can be identified:

- *unification functions* which map data from 2^Φ to Φ. They can be oriented to statistical indexes, such as means or deviations. Of special interest is the function *restrictive unification* defined for all types of data as:

$$f(\{d_1,\ldots,d_n\}) = \begin{cases} d, & \text{if } d_1 = \ldots = d_n \\ undefined, & \text{otherwise} \end{cases}$$

[3] We call a pseudo-instance because ℓ does not belong to the ontology, although in order to improve the clarity of the explanation we will call it instance selector.

[4] For example, if $\Phi^D = \{\mathcal{Z}\}$, $c_{\Phi^D} = \{\langle \mathcal{Z}, rangeOfIntegerSets\rangle\}$, where the function $rangeOfIntegerSets$ has domain $\mathcal{Z}$ and as images the most compact representations of integer sets, $2^{\mathcal{Z}}$, using integer range sets, then $rangeOfIntegerSets(\{1,2,3,4,5,7\}) = [1,5] \cup [7,7]$.

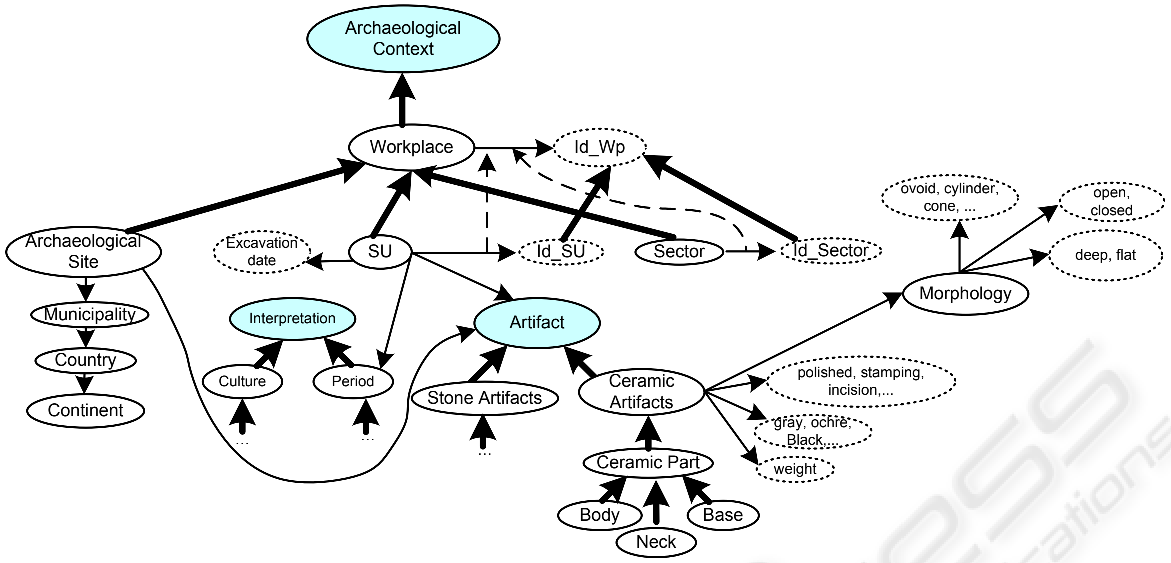

Figure 2: Fragment of an archeology ontology. The concepts are represented in ellipses, the shady ones correspond to root concepts of different hierarchies of the ontology and the dashed-line ones correspond to datatypes. The thick lines represent the hierarchical relations between concepts, the thin lines aggregation relations and the dashed lines represent hierarchies between relations.

- *generalization functions*: which map data from 2^{Φ} to a *compact notation*.

Example 3. Let d be an instance represented by the set $\{$*has_color(a, "black"), has_decoration("incisions"), has_weight(20g)*$\}$ and d' an instance represented by $\{$*has_color(a, "ochre red"), has_decoration("incisions"), has_weight(12g)*$\}$. The outcome of the combination of d and d' by using the union of sets as combination function for the *has_color* and *has_decoration* relations and the maximum of values as combination function for *has_weight* is $\{$*has_color(a,* $\{$*"black", "ochre red"*$\}$*), has_decoration(*$\{$*"incisions"*$\}$*), has_weight(20g)*$\}$. However, if all relations have to be combined using unification functions, the result is undefined, because d and d' have different values for the same relations.

4 MULTIDIMENSIONAL CONCEPTUAL SPACES

The definitions of multidimensional conceptual schema and multidimensional conceptual space are given below. The former can be seen as the structure which defines how to analyze the information. The latter is the container where the analyzed instances are described according to the specifications of the schema.

Definition 9. Let $\mathcal{O}$ be an ontology, $C \in N_C$ and $CC = path_1, ..., path_n$ a set of paths from concept C toward concepts $C_1^*, ..., C_n^*$, respectively, (i.e., $path_i = \oplus_{1 \le j \le n_{i-1}} \left\langle RS_{i_j}, C_{i_j}^* \right\rangle \oplus \left\langle R_{i_{n_i}}, C_n^* \right\rangle$). The tuple $E = (D_1, ..., D_n, c_{\phi_1}, ..., c_{\phi_n})$ is an n-dimensional (or simply multidimensional) conceptual schema of $\mathcal{O}$ associated to C using paths CC, where it is satisfied that$\forall D_i = (Cd_i, \xrightarrow{\sqsubseteq})$, $\forall C_{i_j} \in Cd_i$, $C_i^* \xrightarrow{\sqsubseteq} *C_{i_j}$ and c_{ϕ_i} is a function which assigns a combination function to each simple type of data that can be reached from a concept in Cd_i.

Definition 10. Let $\mathcal{O}$ be a concrete ontology with Abox $\mathcal{A}$. Let E be an m-dimensional conceptual schema of $\mathcal{O}$ associated to C using the paths in CC. The set of tuples $\{t_1, ..., t_n\}$, $t_i = (d_{i_1}, ..., d_{i_m})$, $t_i \neq t_j, \forall i, j \in \{1, ..., n\}$ is called m-dimensional conceptual space of $\mathcal{O}$ with respect to E, if for each data:

1. $d_{i_k} \in C_{i_s}^I, C_{i_s} \in Cd_i$, or
2. $\exists d \in C_{i_s}^I$ such that d_{i_k} is an instance selected with respect to a selector instance of class C_{i_s}, or
3. d_{i_k} is a generalized data of a dataset selected with respect to a selector instance of class $C_{i_s}, C_{i_s} \in Cd_i$.

Each t_i represents a generalized instance of a set of instances of $\mathcal{A}$ selected with respect to a selector instance of class C which contains all paths in CC.

Table 1: Analysis and generalization of ceramic artifacts. Each row maintains the number of instances represented by the associated description (values between parenthesis).

Decoration	Morphology	Weight [g]

incisions	group:{open}
	order:{ovoid, spheroid}
		20 (3)

polished	group:{open, closed}
	order:{cone, ovoid}
		500 (3)

stamping	group:{open}
	order:{cone}
		5 (2)

without dec.	group:{closed}
	order:{spheroid}
		12 (2)

Example 4. The multidimensional schema shown in Table 1 has been constructed by using the following multidimensional conceptual space associated to the concept $CeramicArtifact$, $E = (D_1, D_2, D_3, c_{\phi_{union}}, c_{\phi_{union}}, c_{\phi_{max}})$, where:

$$\begin{aligned} &D_1 = (\{Decoration\}, \emptyset),\\ &D_2 = (\{Morphology\}, \emptyset),\\ &D_3 = (\{Weight\}, \emptyset),\\ &c_{\phi_{union}} = \{\langle \Phi, \hat{\cup} \rangle \,|\Phi \in \Phi^D\},\\ &c_{\phi_{max}} = \{\langle \Re, Max \rangle\}. \end{aligned}$$

$\hat{\cup}$ is defined by $\hat{\cup}(d_1, ..., d_n) = \{d_1\} \cup ... \cup \{d_n\}$ and Max represent the maximum function for real numbers (in this case to compute the maximum weight in each generalization).

Algorithm for the generation of multidimensional conceptual spaces

Algorithm 1 describes how an m-dimensional conceptual space is obtained from a given ontology $\mathcal{O}$ and a conceptual schema of $\mathcal{O}$ associated to the class C, E. The technique of attribute-oriented induction (Carter and Hamilton, 1998; Han et al., 1998) was taken as inspiration for its simplicity and flexibility. One remarkable common feature between such technique and this one is that no restrictions are put on the data. The first step of the algorithm is to define a mapping between each data of each dimension and its generalized value, according to the conceptual schema for instances of type C. Then, the generalized instances are formed, substituting each m-tuple with a generalized m-tuple that constitutes the generalization of the instances of the same type for each dimension.

Algorithm 1 Generation of multidimensional conceptual spaces.

```
Require: $\mathcal{O}, E, C, \beta_m, \beta_M$
  { $\mathcal{O}$, instantiated ontology with Abox $\mathcal{A}$,
  $C$, reference class to generate multidimensional spaces,
  $E$, multidimensional schema,
  $\beta_m$, $\beta_M$, minimum and maximum percentage of different values in each dimension}
Ensure: $E'$
  {$E'$, multidimensional space associated to the schema $E$ and the ontology $\mathcal{O}$ }

  First part: Eliminate the irrelevant dimensions and create the generalization mappings between the data.
  $\mathcal{A}_C = \{a|a \in C^I\}$.
  Iterate $\mathcal{A}_C$ and group different data associated to each dimension of $E$.
  if $\beta_m \geq \frac{|D_i|}{|\mathcal{A}_C|} \geq \beta_M$ ($D_i$ must be removed) then
    $E = (D_1, ..., D_{i-1}, D_{i+1}, ..., D_n, c_{\phi_1}, ..., c_{\phi_{i-1}}, c_{\phi_{i+1}}, ..., c_{\phi_n})$
  else
    Generate mappings $(d, d')$ for each value $d$ collected in $D_i$, $d(d \downarrow_{C_i^{sup}} d')$ with $C_i^{sup}$ being one of the classes direct ancestors of $C_i$, $d \in C_i^I$.
  Second part: Creation of the multidimensional space
  $E' = \{\}$
  for all $a \in \mathcal{A}_C$ do
    Let $t = (d_1, ..., d_n)$ be the tuple of data associated to $a$, according to $E$.
    $t' = (d_1^{'}, ..., d_n^{'})$, computed from the mapping generated in the previous step, being $C_i^{'}$ the type of data $d_i$.
    if $\exists t'' = (d_1^{''}, ..., d_n^{''}) \in E'$, being $C_i^{''}$ the type of data $d_i^{''}$ such that $\forall i \in \{1, .., n\}$ then
      $t'' = (d_1^{''} \uplus_t d_1^{'}, ..., d_n^{''} \uplus_t d_n^{'})$
    else
      $E' = E' \cup \{t'\}$
```

5 INTERESTING CONCEPTUAL SPACES

A conceptual space as previously defined allows users to freely browse the conglomerate of objects and review the aspects they consider more interesting. If the data analyst were not informed on the features of the object distribution in the domain, or if the number of such features were too high, her analysis capabilities would be strongly affected. Nevertheless, this problem can be overcome with an analysis tool able to suggest to the user some interesting analysis dimensions. This can be obtained with a customized *feature selection* process. The concept of feature selection was introduced for the task of dimensionality reduction originally defined in Statistics and widely studied in Machine Learning.

Anyway, it is necessary to define a way of assessing the importance of a given feature subset. The measures more used in the literature are the information gain , the Gini index , the uncertainty and the correlation coefficients. Nevertheless, the large number of studies that argue in favor of decision trees and information gain (like ID3 and C4.5), made us decide to choose such a combination for our feature selection process. More exactly, in this work we propose

Let Cp be an ancestor concept in a dimension, $\{C_1, \dots C_n\}$ concepts directly specialized of Cp, and $objs$ a function which associates each concept with its objects set. Let β be the percentage of maximum correlation between the number of objects of descendant and ancestor concepts.

```
If $\exists C_i$ such that $|objs(C_i)| \geq \beta |objs(C_p)|$
    Promote concepts $C_1, \dots, C_n$ to level of concept $C_p$
    Delete $C_p$ from the dimension.
```

Figure 3: Rules for filtering out uninteresting concepts.

to compute interesting conceptual multidimensional schemata associated to a concept C by way of algorithm 2, an adaptation of the one proposed by (Han and Kamber, 2001). The purpose of such customized algorithm is that of using the distributions of a set of concepts in relation to a set of classes, in order to select the compositions of relations (paths) that assure the highest information gain with respect to the distribution. The main block of the algorithm is procedure *ComputePseudoSchemata* which selects, as a first step, the paths with highest information gain. Then, for each path, the initial distribution is subdivided according to the possible values of the data associated to the objects through such path, and the process of subdivision of the clusters is repeated while the information gain is maintained in a desirable range. A parallel task performed during this process is the computation of the weights which indicate the interest estimation for each conceptual schema that may be generated for each path set.

A further filtering step can be done for each dimension taking into account the relation between the quantity of objects associated to a concept and to its ancestor concept. In this way, an uninteresting ancestor concept can be removed from the dimension, following the rule described in Figure 3.

Algorithm 2 Generation of interesting schemata of multidimensional conceptual spaces.

```
Require: $\mathcal{O}, C, I, \gamma, \alpha$
  {$\mathcal{O}$, ontology with Abox A $I$
  $C$, reference class for generating a schema of multidimensional spaces,
  $\gamma$, minimum allowed information gain
  $\alpha$, minimum number of objects in a description}
Ensure: $SP$
  {$SP$, set of paths associated to $C$ whose subsets can be used to form interesting multidimensional schemata}

  Let $Paths_C$ be the dictionary of paths starting from $C$ with key in the destination concept.
    $S = \{S_i | S_i = \{a \in C_i^I\}, \forall C_i, i \in \{1, \dots, n\}, C \sqsubseteq C_i,$
        $C_i \neq C_j, j \in \{1, \dots, n\}, i \neq j\}$
  $SP = ComputePseudoSchemata(Paths_C, S, \gamma)$

  function $ComputePseudoSchemata(Paths_C, S, \gamma)$ :
  {Outputs a pair, in which the first element is a set of paths and the second a real value indicating the importance of the set of paths for the generation of interesting schemata}
  First phase: Compute the importance of the current clustering
  $ve = \sum_{S_i \in S} imp(S_i)$, where
      $imp(S_i) = \begin{cases} 1, & \text{if } |S_i| > \alpha \\ 1 - |S_i|/\alpha, & \text{otherwise} \end{cases}$
  Second phase: Retrieve the subsets with highest information gains
  if $|S| = 1$ then
    Output $(\emptyset, ve)$
  else
    $SP = \emptyset$
    Compute information gain, $G$, for each destination concept $Paths_C$ according to the classification in $S$
    Let $\{path_1, \dots, path_m\}$ be the set of paths which allow a high discrimination between objects ordered according to the gain value: $G(path_1) \geq \dots \geq G(path_n) > \gamma$ and $C_k$ the destination concept associated to path $path_k$, $k \in \{1, \dots, m\}$.
    for all $k \in \{1, \dots, m\}$ do
      $CD = \{C' | C' \sqsubseteq C_k\}$
      for all $C' \in CD$ do
        $S = \{S_{Cl_{C'}} = \{a \in C_i^I | C_i \in \{1, \dots, n\}, C \sqsubseteq C_i$; $a$ is related to some data $d$ according to $path_i \wedge d \in C'^I\}\}$
        $SP = SP \cup_{(sp,v) \in ComputePseudoSchemata(Paths_C - \{path_i\}, S, \gamma)} \{\langle C_i \cup sp, v + ve \rangle\}$
    Output $SP$
```

6 USING A MULTIDIMENSIONAL CONCEPTUAL SPACE

As explained in the introduction, the major advantage of a multidimensional space is that a user can see her data from different points of view. Tabular models in 3D, function graphs, histograms and relational graphs are the most natural tools to use for the analysis of results. The possibility of realizing generalizations and selections at each level also represents a powerful analysis skill. In this way, it is possible to characterize object classes in relation to others, allowing for the comparison and discovering of class features.

Although these are the analysis methods that have traditionally been used, an analyst may be interested in other more complex insights about the behavior of her data. Various pattern analysis tools have been described in the literature, especially with the development of data mining research. It is thus plausible to create new algorithms for the extraction of interesting patterns in the multidimensional conceptual environment (Han and Kamber, 2001). Some of the most interesting patterns to extract are:

- *of characterization*: they represent rules for characterizing a class of objects according to the values of a subset of its dimensions. They can be expressed by: $class\ X \Rightarrow Condition[p_c]$, where $p_c = \frac{100 \times count(Condition)}{n}$, which means that, in class X, $Condition$ occurs in a p_c percentage of the cases, *Count* is a function that counts the number of times in which a certain condition occurs, and n is the total numbers of analyzed objects. p_c is known as characterization coefficient.
- *of discrimination*: they represent rules for characterizing a class of objects for which a given pattern is not observable with a certain frequency in any other class. They can be expressed by: $class\ X \Leftarrow Condition[p_d]$, where $p_d = \frac{100 \times count(Condition \wedge classX)}{count(Condition)}$. p_d is known as discrimination coefficient.
- *association rules at different levels*: they represent rules for characterizing a class of objects in which co-occurrence relations can be found in the attributes of the multidimensional space. They can be expressed by: $Condition_1 \Rightarrow Condition_2[s,c]$, where $s = \frac{100 \times count(Condition_1 \wedge Condition_2)}{n}, c = \frac{100 \times count(Condition_1)}{count(Condition_1 \wedge Condition_2)}$, which means that in a s% of the objects both $Condition_1$ and $Condition_2$ are observed and that c% of objects satisfying $Condition_1$ also satisfy $Condition_2$. s is known as the support of the rule and c as its confidence.

In order to customize these results to the model we presented, the multidimensional conceptual model must take into account how many objects in the concrete ontology are characterized by the description of each cell.

7 CONCLUSION

The proposal of this paper is the formalization of a data warehouse tool for the Semantic Web. The tool is based on the theoretical foundations of Description Logic and on the current developments of information data generalization. Besides, an algorithm to generate interesting conceptual spaces according to the data distribution is proposed. Ideas for adapting Data Mining techniques in order to allow users a better knowledge discovering from the Semantic Web have also been exposed. Implementation of the proposal framework, on which we are now working, consists of two main components: 1) a reasoner (which works in an off-line way) that retrieve instance models and abstraction functions from an ontology; and 2) a data warehouse processor that use such models and functions in order to perform all the necessary generalizations. This second module has been optimized considering some of the OLAP solutions.

ACKNOWLEDGEMENTS

This research has been partially supported by the project TIN2005-09098-C05-04 (2006-2008).

REFERENCES

Abelló, A. (2002). *YAM^2: A Multidimensional Conceptual Model*. PhD thesis, Universitat Politécnica de Catalunya.

Baader, F. and Sattler, U. (2003). Description logics with aggregates and concrete domains. *Inf. Syst.*, 28(8):979–1004.

Binh, N. T. and Tjoa, A. M. (2001). Conceptual multidimensional data model based on object-oriented metacube. In *SAC '01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 295–300. ACM Press.

Buzydlowski, J. W., Song, I.-Y., and Hassell, L. (1998). A framework for object-oriented on-line analytic processing. In *DOLAP '98: Proceedings of the 1st ACM international workshop on Data warehousing and OLAP*, pages 10–15. ACM Press.

Carter, C. L. and Hamilton, H. J. (1998). Efficient attribute-oriented generalization for knowledge discovery from large databases. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):193–208.

Danger, R. (2007). *Extracción y análisis de información desde la perspectiva de la Web Semántica (Information extraction and analysis from the viewpoint of Semantic Web, in spanish)*. PhD thesis, Universitat Jaime I.

Hacid, M.-S. and Sattler, U. (1998). Modeling multidimensional databases: A formal object-centered approach. In *Proceedings of the Sixth European Conference on Information Systems*.

Han, J. and Kamber, M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Han, J., Nishio, S., Kawano, H., and Wang, W. (1998). Generalization-based data mining in object-oriented databases using an object cube model. *Data Knowledge Engineering*, 25(1-2):55–97.

Nguyen, T. B., Tjoa, A. M., and Wagner, R. (2000). An object oriented multidimensional data model for OLAP. In *Web-Age Information Management*, pages 69–82.

Trujillo, J., Palomar, M., Gómez, J., and Song, I.-Y. (2001). Designing data warehouses with OO conceptual models. *Computer*, 34(12):66–75.