Instance-based domain ontological view creation

Xue, Y.; Ghenniwa, H.; Shen, W.

Instance-based domain ontological view creation

NRCC-51130

Xue, Y.; Ghenniwa, H.; Shen, W.

April 2009

A version of this document is published in / Une version de ce document se trouve dans:
The 2009 13th IEEE International Conference on Computer Supported Cooperative Work in Design, Santiago, Chile, April 22-24, 2009, pp.1-7

Instance-based Domain Ontological View Creation

Yunjiao Xue [1], Hamada H. Ghenniwa [1], Weiming Shen [1, 2]
[1] *Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada*
[2] *Centre for Computer-assisted Construction Technologies, National Research Council, London, Ontario, Canada*
yxue24@uwo.ca, hghenniwa@eng.uwo.ca, wshen@uwo.ca

Abstract

Today in many domains there are very limited explicit ontologies established for building information systems. The information systems have only schemas for their information repositories which to some extent imply the semantics of the information. Traditional ontology-driven semantic integration approaches cannot be directly applied in integrating these information systems. In our work we use the schemas and data instances of the information repositories to discover semantic correspondences between the schema elements and build a domain ontological view. We apply the hierarchical clustering technique on the data instances and use the clusters in the further analysis to reduce the cost of processing a large amount of data. The matching of schema elements is based on the probability distribution of the data instances. The preliminary results have demonstrated the effectiveness of this approach.

Keywords: Instance-based, Ontological View, Schema, Clustering, Probability Distribution, Semantic Integration.

1. Introduction

In an open, dynamic, and distributed environment, various computer systems, such as different collaborative design and manufacturing systems [9], need to collaborate to support information exchange and other requirements. Each computer system is usually a combination of a set of software applications that provides services based on one or more information repositories which have structured and formally represented schemas. A schema is a formal declarative model representing a set of real-world objects, usually within a database.

Due to the nature of being independently designed and built, the computer systems, even for the same domain, are often heterogeneous in terms of the supporting infrastructure (hardware, operating systems, communication facilities, etc), syntactic representation of information, schematic design of information repositories, and semantics of information, which will significantly hinder the collaboration between these systems. There are already mature solutions for the first three issues. The final issue, also known as the semantic integration problem [14], is attracting more and more attention from today's research communities.

Ontology-driven semantic integration is one of the solutions for the semantic integration problem [19]. This solution is based on available ontologies. However, in many domains, the case is that there are no pre-established explicit ontologies and the information semantics are embedded in the code of the applications and the underlying information repositories. For example, a database-based information management system works on a database schema. A schema is not a formal ontology but to some extent it implies the semantics of the information that it manages. For example, a relational database schema contains multiple table definitions and each table can represent a concept. Accordingly, data rows in a table represent instances of the concept. Each schema actually reflects a specific conceptual view of the domain and can represent an *ontological view*. The formal definition of ontological view will be given in section 3.

This background motivates us to consider a new approach to perform semantic integration upon a set of information systems in a domain without explicit ontologies with a two-stage process:

(1) Pick up a subset of the information systems, elicit the local ontological views that the systems represent, and create a domain ontological view (or global ontological view) based on the local ontological views of these information systems. The objective of creating the domain ontological view is to establish a conceptual model for the domain. Note that the domain ontological view is also only a view of the domain conceptualization, which may be more complete and commonly agreeable than each local ontological view but still not the "domain ontology". The principle of creating the domain ontological view is: if several schema elements from multiple information systems are discovered to be semantically equal to each other, then they refer to the same concept and therefore, a concept which stands for these schema elements should be contained in the domain ontological view. The ontology

can be specified with some ontology language such as OWL [11].

(2) This domain ontological view (as a conceptual model for the domain) can be used in further semantic integration, i.e., matching the concepts in the domain ontological view to the schema elements of other information systems (or new systems that are currently unknown). The result of the matching is a set of relationships between the concepts in the domain ontological view and the schema elements, meaning that these schema elements are semantically equivalent to the concepts. For example, if it is identified that concept C in the domain ontological view O is equivalent to schema elements e_1, e_2, ..., e_m from various information systems, it can be concluded that e_1, e_2, ..., e_m are representing the same concept, therefore it is possible to exchange information representing the same concept among these systems.

In the first stage, the way that data instances are maintained in the information repositories can be applied to increase the precision of discovering equivalence relationship between schema elements (such as two tables) since the data instances can provide plenty of useful clues. Simply using all the data instances to identify the equivalence relationship may cause performance issues due to the amount of data. It is necessary to find a cost-sensitive approach to reduce the cost of performing the equivalence relationship discovery.

This paper addresses these issues based on our research on ontologies and semantic integrations. In section 2, we analyze some related work. Section 3 provides the theoretical foundation for the ontology theory. In this section we introduce the term "ontological view" which illustrates the nature of conceptualizations in a better way than the term "ontology". Section 4 formulates the problem. The solutions are presented in section 5. We present the preliminary results in section 6 and conclude our work in section 7.

2. Related Work

Ontology plays an important role in understanding and dealing with the information semantics. An ontology is a formal and explicit specification of a conceptualization [17].

Simply, an ontology specifies the concepts and relationships between the concepts in a domain [8]. In other words, we say that we know a concept if it is conceptualized in an ontology. The ontology theory establishes an assumption: if an ontology is formally and explicitly represented in a machine readable and processable way, we say that the computers "know" the concepts within the ontology. Having the ontology provided, the computers can act as if they understand the concepts. For example, given an ontology represented by an OWL file and a concept "Human" in the ontology (a string in a predicate in the file), we say that the computer and application system that have access to the ontology can understand the concept "Human". In other words, if an application system can associate an information item to the concept "Human", it is able to process the information item in a way that semantically reflects the concept's meaning. If multiple systems know "Human", together they can collaborate to process information related to this concept.

The general idea of ontology-based semantic integration approaches includes the following aspects [10]:

(1) One or more explicit ontologies are provided for a domain of discourse. The ontologies may be different but they commit to the same domain.

(2) The computer systems in this domain are built based on the ontologies that specify the conceptualizations of the domain.

(3) In the case of multiple ontologies, existing, ontology integration can be performed to discover semantic correspondences among various ontologies, and a more complete domain ontology can be created (if necessary) based on the individual ontologies and semantic correspondences among them.

(4) Using the semantic correspondences among ontologies, the information can be transferred from one system to another (with necessary transformation) or integrated in a semantically correct way.

In many cases, the application of ontology-based approaches is limited due to the lack of explicit ontologies. Instead, since schemas are usually available in many information systems, schema-based approaches play an important role in information integration.

Schema matching is an important research topic which aims at finding semantic equivalence relationships between schema elements such as database tables and the table columns. Bohannon et al. [12] view schema matching as a pairing of attributes (or groups of attributes) from the source schema and attributes of the target schema such that the pairs are likely to be semantically related. Since the computation of schema matching usually involves many data tables and attributes which significantly increase the workload, automated support for schema matching has received a great deal of attention in the research community. A recent complete survey can be found in [3]. Schema matches can be discovered by analyzing the similarity of schema information, preservation of constraints, domain knowledge, and instance data. The results of the automatic analysis are candidates of the possible matches, which still need to be verified by human experts.

In the family of schema matching approaches, instance-based approaches [3] can make use of the data instances which imply lots of clues for the potential attribute matches. One of the major issues of these approaches is the cost of manipulating a large quantity of raw data. A solution to increase the efficiency is to use instance representatives (with each representing a set of data instances) for the analysis instead of using all

raw data. The clustering methods can be applied to this solution.

Some researches also use the clustering methods to find closely related schema elements. For example, Pei et al. [2] proposed a new approach for schema matching by clustering schemas on the basis of their contextual similarity and clustering attributes of the schemas that are in the same schema cluster to find attribute correspondences between these schemas. The approach also clusters attributes across different schema clusters using statistical information gleaned from the exiting attribute clusters to find attribute correspondences between more schemas. Smiljanic et al. [18] presented a clustering based technique for improving the efficiency of XML schema matching by partitioning schemas with clusters and reducing the overall matching load. In this work clustering is used to quickly identify regions, i.e., clusters, in the large schema repository which are likely to produce good mapping. These researches have a different context than our work, i.e., they cluster the schema elements instead of clustering the data instances.

3. Ontological View

As mentioned in [17], an ontology is a formal and explicit specification of a conceptualization. An ontology itself needs to be specified by a *language*. Therefore, a more complete definition should consider the factor of a language.

A formal definition for o*ntology* is provided in [7]. This definition is based on a language **L** that is used to specify the ontology. A language is composed of a vocabulary and a set of models of the language. A language **L** commits to a conceptualization **C** of a domain **D** by means of an ontological commitment **K**. **K** constrains the intensional interpretation of the **L**, i.e., the language is used in an intended way for a domain instead of an arbitrary way. Given a language **L** and an ontological commitment **K**, the set $\mathbf{I_K}(\mathbf{L})$ of all models of **L** that are compatible with **K** is called the set of *intended models* of **L** according to **K.** Given a language **L** with ontological commitment **K**, an *ontology* for **L** is a set of axioms designed in a way such that the set of its models approximates as best as possible the set of intended models of **L** according to **K**. More detailed definitions for these terms can be found in [7].

Since a domain can be conceptualized in various ways, there is actually not just one unique conceptualization (and therefore not just one unique ontology) for a domain. Instead, different views, with each reflecting a specific view or the domain conceptualization, may exist. Here we present the definitions for *ontological views*. Given a language **L** and another conceptualization **C'**, **L** can commit to **C'** by means of an ontological comment **K'**. Given a language **L**, with ontological commitment of view **K'**, an *ontological view* for **L** is a set of axioms designed in a such a way that the set of its models approximates as best as possible the intended models of **L** according to **K'**.

When different languages are employed, we also see that the axioms designed with each language and its ontological commitment actually compose an *ontological view* for the domain. In the simplest case, if two languages have different vocabularies with different symbols, but some pairs of symbols from two vocabularies are semantically equivalent (i.e., they are synonyms) which implies a partial overlap of their intended models, then the axioms written by the languages compose different ontological views that are possible to be semantically integrated. That is, given a ontological view O with intended model $\mathbf{I_K}(\mathbf{L})$ and another ontological view O' with intended model $\mathbf{I_{K'}}(\mathbf{L'})$, O and O' are integratable (denoted by ◊) if and only if $\mathbf{I_K}(\mathbf{L})$ overlaps with $\mathbf{I_{K'}}(\mathbf{L'})$. That is,

$$(\mathbf{I_K}(\mathbf{L}) \neq \mathbf{I_{K'}}(\mathbf{L'})) \wedge (\mathbf{I_K}(\mathbf{L}) \cap \mathbf{I_{K'}}(\mathbf{L'}) \neq \varnothing) \leftrightarrow (O \lozenge O')$$

This can be illustrated by the following Figure 1:

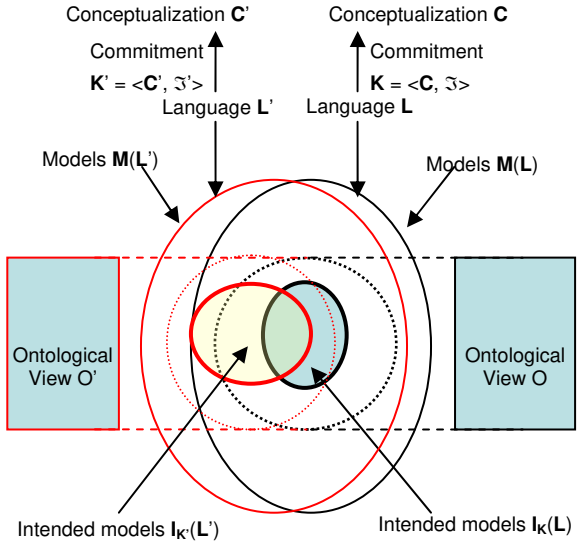

Figure 1. Different ontological views with different languages for different conceptualizations which sets of intended models overlap.

Given a source language $\mathbf{L_S}$ (which vocabulary is V_S) with ontological commitment of view $\mathbf{K_S}$ for conceptualization $\mathbf{C_S}$ and a target language $\mathbf{L_T}$ (which vocabulary is V_T) with ontological commitment of view $\mathbf{K_S}$ for conceptualization $\mathbf{C_T}$, an *ontological equivalence mapping* is a function from V_S to V_T, m: $V_S \rightarrow V_T$ assigning symbols in V_T to the ones in V_S which share the same intensional interpretation, i.e., i) for constant symbols $c_S \in V_S$ and $c_T \in V_T$, $m(c_S) = c_T$ if and only if i) there exists a common concept d in C_S and C_T, both c_S and c_T are interpreted as d; ii) for predicate symbols $p_S \in V_S$ and $p_T \in V_T$, $m(p_S) = p_T$ if and only if there exists a common conceptual relation ρ in $\mathbf{C_S}$ and $\mathbf{C_T}$, both p_S and p_T are interpreted as ρ. It is obvious that an important task in semantic integration is to discover the

ontological equivalence mapping between two ontological views.

In each information system, the schema is specified with a selected language with specific interpretation in the domain. Even by rigid definition a schema is not an ontology. In a sense, it reveals a specific view of the domain conceptualization so it can be viewed (with necessary transformation) as an ontological view. Based on these schemas, another ontological view (which may be more complete but is still not the actual domain ontology) can be built with a specific language.

4. Formulating the Problem

We adopt the similar ideas in *schema matching* to formulate the problem. We try to discover semantic relationships between the elements, mainly the concepts and the equivalence relationship, of multiple ontological views. Since the schemas may use different modeling paradigms and languages, they need to be converted to a unified paradigm such as the FRAME model [1] first. Each FRAME model represents an ontological view. The following Figure 2 illustrates this idea.

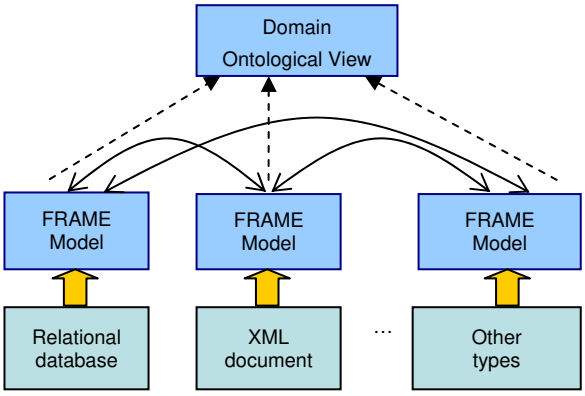

Figure 2. Creation of domain ontological view.

Given an ontological view O_1 with a set of concepts $C_1 = \{c_{11}, c_{12}, \ldots, c_{1n}\}$ and another ontological view O_2 with a set of concepts $C_2 = \{c_{21}, c_{22}, \ldots, i_{2m}\}$, the goal of ontological view matching is to discover concept mappings, i.e., pairs of matching concepts c_{1i} and c_{2j} such that c_{1i} and c_{2j} represent the same real world concept, $1 \le i \le n$, $1 \le j \le m$. We denote a concept mapping with c_{1i} a c_{2j}. When put together, they compose an ontological view mapping $m = \{ c_{1i}$ a $c_{2j} \mid c_{1i} \in C_1, c_{2j} \in C_2 \}$.

Now we look into the concepts. Each concept c can be modeled (or specified by) as a set of *properties*, i.e., $c = \{p_1, p_2, \ldots, p_n\}$, where each p_i is a property, $1 \le i \le n$. We rely on the assumption that the similarity of the representations of properties, i.e., the syntactic similarity of concept properties, indicates the semantic similarity of real-world objects modeled with these concepts. For example, for two concepts $c_1 = \{p_{11}, p_{12}, \ldots, p_{1n}\}$ and $c_2 = \{p_{21}, p_{22}, \ldots, p_{2m}\}$ from two ontological views, if most of their properties can be discovered as similar, e. g., $p_{11} \approx p_{21}$ ($\approx$ denotes semantically similar), $p_{12} \approx p_{22}, \ldots, p_{1k} \approx p_{2k}$, $k \le \min\{n, m\}$ and k is a large enough number, then it can be claimed that c_1 and c_2 are semantically equivalent (referring to the same real-world concept).

Next, we present the solutions that adopt data instances to discover the similarity of properties.

5. Instance-based Semantic Relationship Discovery

Instance values can provide useful clues to help discover the similarity of concept properties. The probability distribution (or probability density) is one of the often-used approaches to analyze the instance values. If two properties of two concepts have compatible data types (the data type can be known from the schema) and the probability distributions of their instance values are identical or very close, then it is reasonable to infer that these two properties are very likely to be semantically similar. The problem here is:

- How one can estimate a probability density function $f(x)$ given a sequence of independent and identically distributed random variables $x_1, x_2, \ldots, x_n$ from this density f?

There is a rich collection of non-parametric density estimators, including kernel, spline, orthogonal, series, and histogram [6].

We use Kernel density estimation [4, 5] to compute the probability distribution of the instance values. In statistics, Kernel density estimation is a non-parametric way of estimating the probability density function of a random variable. Different than many distributions, the Kernel density estimation is smooth and independent of end points. It just depends on the bandwidth.

The definition of kernel density estimation is presented as follows.

If $x_1, x_2, \ldots, x_N \sim f$ is an independent and identically-distributed random variables sample of a random variable, then the kernel density approximation of its probability density function is

$$\hat{f}_h(x) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x - x_i}{h}) \quad ,$$ where K is some kernel and h is the bandwidth (smoothing parameter). Quite often K is taken to be a standard Gaussian function with mean zero and variance 1:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad .$$

After we get the probability densities of the properties, we need to compare them and check their similarity. Here another question is raised:

- How to compare different probability densities?

We employ the Kullback-Leibler (K-L) divergence approach [15] to compare the probability densities. In probability theory and information theory, the K-L divergence (also named information divergence, information gain, or relative entropy) is a non-commutative measure of the difference between two probability densities.

The definition of K-L divergence is presented as follows:

For probability densities f_1 and f_2 of a continuous random variable, the K-L divergence of them is defined as

$$\delta(f_1, f_2) = \int_{-\infty}^{\infty} f_1(x) \log \frac{f_1(x)}{f_2(x)} dx .$$

The K-L divergence can be considered as a kind of a distance between the two probability densities, though it is not a real distance measure because it is not symmetric.

In the instance-based analysis, another issue is when using original data instances to compute the probability densities and compare them, the computation cost is very high due to the large amount of the raw data. The solution is to utilize the DBMS's capability of managing data efficiently and cluster the data first, and then compute the probability densities based on the clustered data.

Cluster analysis [13], also called data segmentation, relates to grouping or segmenting a collection of objects (also called observations, individuals, cases, or data rows) into subsets or "clusters" such that those within each cluster are more closely related to one another than objects assigned to different clusters. Since objects in each cluster are closer or similar to each other, it is reasonable to use one typical object within one cluster to represent the entire cluster. The typical object is a weighted cluster center which can represent a set of values similar to the center itself. The use of a typical object will significantly reduce the size of the problem.

Hierarchical clustering is one of the major methods of cluster analysis. In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions take place, which may run from a single cluster containing all objects to *n* clusters with each containing a single object. Hierarchical clustering is subdivided into agglomerative methods, which proceed by a series of fusions of the *n* objects into clusters, and divisive methods, which separate *n* objects successively into finer clusters. A key component of the analysis is repeated calculation of distance measures between objects, and between clusters once objects begin to be grouped into clusters.

The initial data for the hierarchical clustering of N objects is a set of $\frac{N \times (N-1)}{2}$ object-to-object distances and a *linkage function* for computation of the cluster-to-cluster distances. The linkage function is an essential prerequisite for hierarchical clustering. Its value is a measure of the *distance* between two groups of objects, i.e. two clusters.

A commonly used linkage function is *complete linkage clustering*, in which distance between groups is defined as that of the furthest pair of individuals, where a pair consists of one member from each cluster. Mathematically, the complete linkage function—the distance $D(X, Y)$ between clusters X and Y—is described by the following expression:

$D(X, Y) = \max(d(x, y)),\ x \in X$ and $y \in Y$

where

- $d(x, y)$ is the distance between elements $x \in X$ and $y \in Y$;
- X and Y are two sets of elements (two clusters).

Complete linkage clustering is an agglomerative method. It starts from the clusters initially containing one element each and successively fuses them to generate larger clusters. Therefore, the two clusters with the lowest distance are joined together to form the new cluster. At each step, the clusters to be used are those that are, according to some pre-defined metric, most similar to each other.

The above discussion shows that the distance between elements is the foundation of cluster analysis. An important work in any clustering is to select an appropriate distance measure, which will determine how the similarity of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another, according to one distance and further away according to another.

At the information level, we consider generic metric space, not definitely pure Euclidean Space (i.e., it is only required that the distance between any pair of elements is known. It is not limited to the coordinates of points). A metric on a set X is a function (called the distance function or simply distance)

$d: X \times X \rightarrow R$,

where R is the set of real numbers. For all x, y, z in X, this function is required to satisfy the following conditions:

(1) $d(x, y) \geq 0$ (non-negativity)

(2) $d(x, y) = 0$ if and only if $x = y$ (identity of indiscemibles). Condition (1) and (2) together produce positive definiteness.

(3) $d(x, y) = d(y, x)$ (symmetry)

(4) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

Since in an information system we usually face three types of data: numeric data, date-time, and text string, we define the distance metric for the three types:

If x, y are values of concept instances on property X, the distance between x and y, $d(x, y)$, is defined as:

- Euclidean distance in Euclidean one dimensional space, $|x - y|$, if the type of X is numeric;
- Euclidean distance in Euclidean one

dimensional space, |absolute_time(x) – absolute_time(y) |, if the type of X is datetime, where absolute_time is a function to map each date time to a long integer;

- Edit distance of string, if the type of X is text string. The edit distance $d(x, y)$ is the minimal cost for a sequence of edit operations to transform x to y.
  The edit operations include:
  (1) Replace one character in x by a character from y;
  (2) Delete one character from x,
  (3) Insert one character from y.
  The cost model is defined as::

$$c(a,b) = \begin{cases} 1, \text{ if } a \neq b \\ 0, \text{ if } a = b \end{cases}$$

a and b can be ε (null character) meaning inserting a new character b or deleting an existing character a.

After the clusters are created, we expect to use the representative data instance in each cluster, i.e. the cluster center, to represent the entire set of data instances in the following analysis. This is known as a 1-median problem [16] which is defined as follows:

Given a universe U, a finite multi-set of points P, and a metric d, a 1-median is a point $m \in U$ that minimizes the objective function

$$\sum_{p \in P} d(p,m)$$

In this definition, m is a valid member in U but not definitely a point in P. Since the median point is relatively closer to other points (in terms of the selected distance metric), it is an optimal one to represent others.

The basic idea of the algorithm of finding the 1-median point is: for a point $p \in P$, let $S(p) = \sum_{x \in P} d(p,x)$, then conduct a series of comparisons between $S(p)$, $p \in P$ to find the a point q that minimizes the value of S. The point q is the cluster center under the i-median.

6. Preliminary Results

The solutions proposed in this paper have been applied to semantically integrating several information systems (with each providing a local ontological view) in the collaborative promotion domain. The systems share some common concepts that are represented as tables in databases (instances of concepts as rows in tables). Since the information is not centralized in one location, these systems need to collaborate to provide information together and support decision making.

The prototype system makes use of the data instances to discover the semantically equivalent elements from each ontological view. A global ontological view is built based on the equivalence relationships between local ontological views. Then, the global ontological view can be used to shoot cross-system query. The efficiency is significantly increased by using the data clusters, especially when there are more than 100, 000 records in the tables.

7. Conclusion and Future Work

Common agreement upon a domain is very important for information systems that need to collaborate to achieve some targets. Ontology can be used to specify domain conceptualization but the fact is, in many domains there are no pre-defined explicit ontologies. In our work we propose the concept of ontological view that can be reflected by the schema in each system. To semantically integrate various systems we need to discover the semantic relationships between the schema elements and build a global ontological view. We propose to use instance-based approaches to discover such relationships. To reduce the cost of computation, we apply the clustering analysis to use representative data instances only.

Our future work will focus on applying and evaluating other approaches for density estimation, probability density comparison, clustering as well as richer collection of linkage functions and distance metrics. More sophisticated evaluation engine combining multiple approaches will also be proposed to improve the discovery results. Furthermore, semantic relationship types, other than the equivalence relationships, such as generalization or specialization, will also be taken into consideration to discover more complete relationships between ontological views, which are able to help improve the quality of the created domain ontological view.

References

[1] M. Minsky, "A Framework for Representing Knowledge", P. Winston (Ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill, 1975, 211-277.

[2] J. Pei, J. Hong and D. Bell, "A Novel Clustering-Based Approach to Schema Matching", *Advances in Information Systems (Lecture Notes in Computer Science, Volume 4243)*, Springer Berlin/Heidelberg, 2006, 60-69.

[3] E. Rahm and P. A. Bernstein, "A Survey of Approaches to Automatic Schema Matching", *The International Journal on Very Large Databases (VLDB)*, 2001, 10(4), 334-350.

[4] L. Wasserman, "*All of Statistics: A Concise Course in Statistical Inference"*, Springer Texts in Statistics, 2005.

[5] A. B. Turlach, "Bandwidth Selection in Kernel Density Estimation: A Review", *CORE and Institut de Statistique*, 1993, 23-493.

[6] S. J. Bean and C. P. Tsokos, "Developments in Nonparametric Density Estimation", *International Statistical Review*, 1980, 48, 267-287.

[7] N. Guarino, "Formal Ontology and Information Systems", *Proceedings of FOIS'98*, Trento, Italy, June 6-8, 1998, Amsterdam, IOS Press, pp. 3-15.

[8] M. Crubzy, Z. Pincus and M. A. Musen, "Mediating Knowledge between Application Components", *Proceedings of the Semantic Integration Workshop of the Second International Semantic Web Conference (ISWC-03)*, Sanibel Island, Florida, 2003.

[9] W. Shen, D. H. Norrie and J. A. Barthes, "*Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing"*, Taylor & Francis, 2001.

[10] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hubner, "Ontology-based Integration of Information – A Survey of Existing Approaches", *Proceedings of the IJCAI'01 Workshop: Ontologies and Information Sharing*, Seattle, Washington, USA, 2001.

[11] W3C, "OWL: Web Ontology Language", http://www.w3.org/TR/owl-features/.

[12] P. Bohannon, E. Elnahrawy, W. Fan and M. Flaster, "Putting Context into Schema Matching", *Proceedings of VLDB'06*, Seoul, Korea, September 12-15, 2006, pp. 307-318.

[13] S. Kotsiantis and P. Pintelas, "Recent Advances in Clustering: A Brief Survey", *WSEAS Transactions on Information Science and Applications*, 2004, 1(1), 73-81.

[14] F. N. Noy, "Semantic Integration: A Survey of Ontology-based Approaches", *SIGMOD Record*, *Special Issue on Semantic Integration*, 2004, 33(4), 65-70.

[15] S. Kullback, "The Kullback-Leibler Distance", *The American Statistician*, 1987, 41:340-341.

[16] Z. Drezner, J. Thisse and G. O. Wesolowsky, "The Minimaxmin Location Problem", *Journal of Regional Science*, 26:87-101, 1986.

[17] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", *KSL-93-04*, Knowledge Systems Laboratory, Standard University, http://ksl-web.stanford.edu/.

[18] M. Smiljanic, M. van Keulen and W. Jonker, "Using Element Clustering to Increase the Efficiency of XML Schema Matching", *Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, 2006, p. 45.

[19] F. Hakimpour and S. Timpf, "Using Ontologies for Resolution of Semantic Heterogeneity in GIS", *Proceedings 4th AGILE Conference on Geographic Information Science*, Brno, Czech Republic, 2001, pp.385-395.