ISSN: 0828-3494
ISBN: 0-7731-0533-6 (print)
ISBN: 0-7731-0534-4 (on-line)

An Ontology-Based Approach to Data Cleaning

Xin Wang, Howard J. Hamilton, and Yashu Bither
Technical Report CS-2005-05
July, 2005

Department of Computer Science
University of Regina
Regina, Saskatchewan
CANADA S4S 0A2

An Ontology-Based Approach to Data Cleaning

Xin Wang, Howard J. Hamilton, and Yashu Bither
Department of Computer Science
University of Regina
Regina, Saskatchewan
CANADA S4S 0A2
{wangx, hamilton, bither1y} @ cs.uregina.ca

Abstract

This paper describes an ontology-based approach to data cleaning. ***Data cleaning*** is the process of detecting and correcting errors in databases. An ***ontology*** is a formal explicit specification of a shared conceptualization of a domain. Our approach to data cleaning requires a set of ontologies describing the domains represented by the classes and their attributes. Using the ontology-based approach, we are able to clean data of not only syntactic errors but also some classes of semantic errors.

1. Introduction

Data cleaning is the removal of random and systematic errors from data elements through filtering, merging, and translation [6]. It requires the largest fraction of time of all the steps in a knowledge discovery process [2][4][8] [19]. Algorithms for data cleaning rely on increasing the internal consistency of data and its consistency with encoded domain knowledge. However, the domain knowledge usually comes from personal knowledge and experiences. Therefore most of data cleaning is conducted at the data level rather then knowledge level. Our goal is to treat data cleaning as a systematic and cost-effective data improvement procedure with a formal domain knowledge support.

An ***ontology*** is a formal explicit specification of a shared conceptualization of a domain. It represents the concepts and their relations that are relevant for a given domain of discourse. It consists of a representational vocabulary with precise definitions of the meanings of the terms of this vocabulary plus a set of axioms.

In this paper, we purpose an ontology-based data cleaning framework. In the framework, data cleaning requires a set of ontologies describing the domains represented by an ontology representation language. Using the ontology-based approach, we are able to clean data of not only syntactic errors but also some classes of semantic errors.

The remainder of this paper is organized as follows. Section 2 provides background information concerning data cleaning, while Section 3 provides background information on ontologies. Section 4 explains our ontology-based approach to data cleaning. Finally, Section 5 draws conclusions.

2. Data Cleaning

According to one definition [12], ***data cleaning*** is a two step process of detection and then correction of errors in a data set.

For data cleaning in general, three comprehensive systems have been developed: AJAX, the Potter's Wheel and Intelliclean. Let us briefly described the main features of each.

AJAX [2][4] provides a declarative framework for describing data cleaning as a series of transformations to data, including the removal of synonymous records. It extends SQL to describe concepts relevant to data transformation [7] and matching. The problem of synonymous records from multiple sources is referred to as the ***object identity problem***[2]. For example, "John Smith" may be referred as "Smith John" or "J. Smith". The five atomic data transformations that were used in the AJAX framework are mapping, merging, clustering, merging, and SQL view. Several atomic transformations can be combined in a pipeline to form a higher level transformation called a ***complex transformation***.

The ***Potter's Wheel*** [15] is an interactive data cleaning system with tightly integrated steps for performing transformations and detecting discrepancies. Using a spreadsheet-like interface, a user incrementally specifies a series of transformations to clean the data. The main components of the Potter's Wheel architecture are a data source, a transformation engine, an online reorderer to support interactive scrolling and sorting for the user interface, and an automatic discrepancy detector. The transformation engine applies transformations in two situations. First, transformations are applied when records are rendered to the screen. With the spreadsheet user interface, this is done when the user scrolls or jumps to a new scrollbar position. Since the number of rows that can be displayed on screen at a time is small, users perceive transformations as being instantaneous. Secondly, transformations are applied to detect discrepancies in transformed versions of data. By integrating discrepancy detection and transformation, the Potter's Wheel allows users to gradually build a single, complex transformation to clean the data by adding transformations as discrepancies are detected. Users can specify transformations through graphical operations or through examples, and see the effect instantaneously, thereby allowing easy experimentation with different transformations.

Intelliclean [15] applies three steps to clean data. First, it preprocesses data to remove abbreviations and standardize data formats. Second, it applies two synonymous record identification rules (Rule 1 and Rule 2) based on the ***certainty factor*** (CF) to detect and match synonymous records. The certainty factor CF, where $0 < CF \leq 1$, represents the confidence in the rule's effectiveness in identifying true duplicates. Rule 1 and Rule 2 represent CFs of 1 and 0.9, respectively. Intelliclean also uses one merge/purge rule to merge and purge the synonymous records. The merge/purge rule is applied only where the CF is greater than a user defined ***threshold*** (TH). For example, records A and B with $CF = 0.8$, $TH = 0.7$ are merged if $CF > TH$. Finally, Intelliclean interacts with the human user to confirm the sets of synonymous records. To allow the user some control over the matching process, the recall and precision are measured and presented to the user. The

recall is defined as the percentage of synonymous records being selected as synonymous records. The ***precision*** (*P*) is percentage of records identified as synonymous records that are indeed synonymous. High recall is achieved by accepting records with low degree of similarity as synonymous, at the cost of lower precision. High precision is achieved analogously at the cost of lower recall.

Although the three systems integrate different methods and are effective to some datasets, they cannot clean some semantic errors due to lack of domain knowledge support.

3. Ontological Representations

In philosophy, the term "ontology" refers to "the study of what there is, an inventory of what exists" or in other words "the attempt to say what entities exist" [9]. Recently, in Computer Science, researchers have formulated explicit representations of the entities that exist in particular domains of application. These researchers use the term "ontology" to refer to a formal, explicit specification of a shared conceptualization of a domain. It is in this latter sense that we use the term "ontology" in this paper.

An ontology represents the concepts and their relations that are relevant for a given domain of discourse [2]. It consists of a representational vocabulary with precise definitions of the meanings of the terms of this vocabulary plus a set of axioms [5].

An ***ontology language*** is a formal language for representing ontologies. Informally, it can be thought of as having properties similar to programming languages and data definition languages. Several ontology languages have been proposed based on various underlying paradigms such as description logic, first-order logic, frame-based representations, taxonomies, semantic nets, and thesauruses. OWL (Web Ontology Language) [12] is based on a description logic. It is designed for use by applications that need to process the content of web-based information instead of just presenting the information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics [12]. Additionally, OWL is reasonably well supported by existing ontology construction tools. For example, the OWL Plugin [11] is an extension of Protégé-2000[9] with support for OWL.

In our own previous work [18], we described an ontology-based approach to spatial clustering. This work was broadly similar to the present work, since an ontology was used to describe spatial data sets and clustering algorithms. The goal of the ONTO_CLUST system is to perform clustering of spatial data according to general guidelines provided by the user and then assist the user in interpreting the results. In the following section, we introduce a framework to combine an ontology with data cleaning.

4. The OntoClean Framework for Ontology-Based Data Cleaning

In this paper, we propose a framework called ***OntoClean*** for ontology-based data cleaning. We assume that ontologies relevant to the database to be cleaned have already been represented an ontology language. The OntoClean framework provides a template for performing data cleaning using the following steps. First, the data cleaning ontology is represented in a web ontology language. Secondly, the user's goal is translated into queries that perform reasoning on the ontology. Relevant data cleaning algorithms and attribute constraints are selected and instantiated from the ontology with respect to the user's goal. Thirdly, the selected data cleaning algorithm is applied to the selected data set based on the results produced from queries. Finally, the results of the cleaning process are provided to the user along with an explanation of what has been performed.

The advantages of the framework are as follows. First, the user's goal is given at the semantic level. The user does not need to know details about the cleaning algorithm. Secondly, the framework combines static knowledge (in the form of an ontology) with problem-solving methods (for data cleaning). Incorporating domain ontologies and task ontologies in data cleaning algorithms can enhance the quality of the cleaning and the user's knowledge about what type of cleaning was performed. Thirdly, the ontology is represented in OWL, the standard web ontology language, so the whole framework can be extended to data sets in the web environment.

The research process can be seen as aspects of three phases: understanding the problem, understanding the data, and performing data processing [15]. Right now, data cleaning can be regarded as occurring in the third phase, data processing, which purely operates on data. Arguably, the most appropriate cleaning algorithm should be selected after taking into account factors such as the user's goal, relevant domain-specific knowledge, characteristics of the data, and available cleaning algorithms. However, if queries were posed to the user about those factors in an arbitrary manner, it would be confusing. An ontology can provide a systematic way of organizing these factors such that they can contribute to the selection process and an orderly description of this process to the user.

The OntoClean framework is shown in Figure 1. The ***data cleaning ontology*** component is used when identifying the cleaning problem and the relevant data. Within this component, the ***task ontology*** specifies the potential methods that may be suitable for meeting the user's goals, and the ***domain ontology*** includes all classes, instances, and axioms in a specific domain. A domain ontology could be built by users or domain experts, or derived from some existing ontologies.

With the framework, users first give their goals for cleaning. The goals are initially represented in natural language. The goals are translated into the ontology query language and matched with task instances in the task ontology. The goals are also used to search the domain ontology. The results of these queries identify the proper cleaning methods. Based on these results, cleaning is conducted. During the cleaning, domain ontology continues to provide domain knowledge such as attribute restraint for checking invalid values. The cleaning result can be used for statistical analysis or it can be

interpreted using the task ontology and the domain ontology. The final result is returned to the user with understandable explanations.

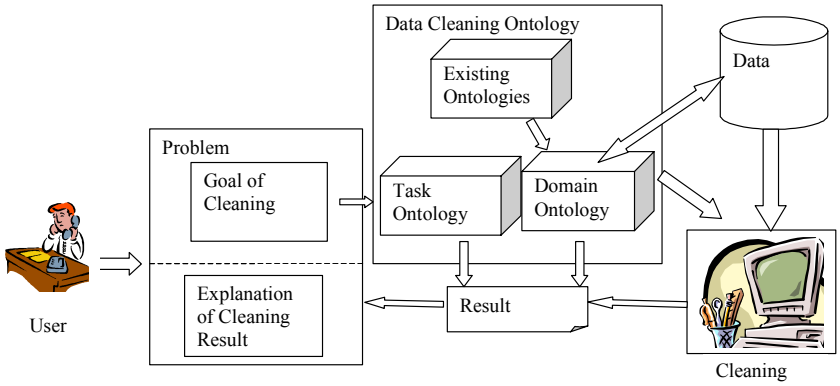

Figure. 1. The OntoClean framework for ontology-based data cleaning

Let us describe three main high-level classes in data cleaning ontology.

1) `DataCleaningTask` is an abstract class. It is the superclass of all possible data cleaning tasks that users may perform, including `CleanSingleDBTask` and `FindOverlapAmongMultiDBTask`. The purpose of the `CleanSingleDBTask` is to detect and correct errors for one single database. It includes some subtask classes such as `StringMatchingTask`, `FindPeculiaritiesDataTask`, `FindInvalidValues`, `FindMissingValues` and `FindSynonymousRecordsTask`. The purpose of the `FindOverlapAmongMultiDBTask` task is check for common attributes or other overlap among multiple databases. The purpose of the `FindSynonymousRecordsTask` task is to find synonymous records, i.e., multiple records (tuples) that represent the same real world entity in different syntactic forms. The most common version of the synonymous record problem occurs when the same person is represented in a contact list with slightly varying names or addresses. Each type of cleaning task is connected to some classes of cleaning algorithms. Based on the purpose of the cleaning and the domain, an appropriate cleaning algorithm is selected.

2) `DataCleaningMethod` represents a list of all available cleaning methods and their features. Every method is connected with some data cleaning tasks. For example, the MatchBox algorithm [2] is an instance of `DataCleaningMethod`. Since the algorithm can be applied to solve the synonymous record problem, the instance is linked with `SynonymousRecordsTask` in ontology (as shown in Figure 2).

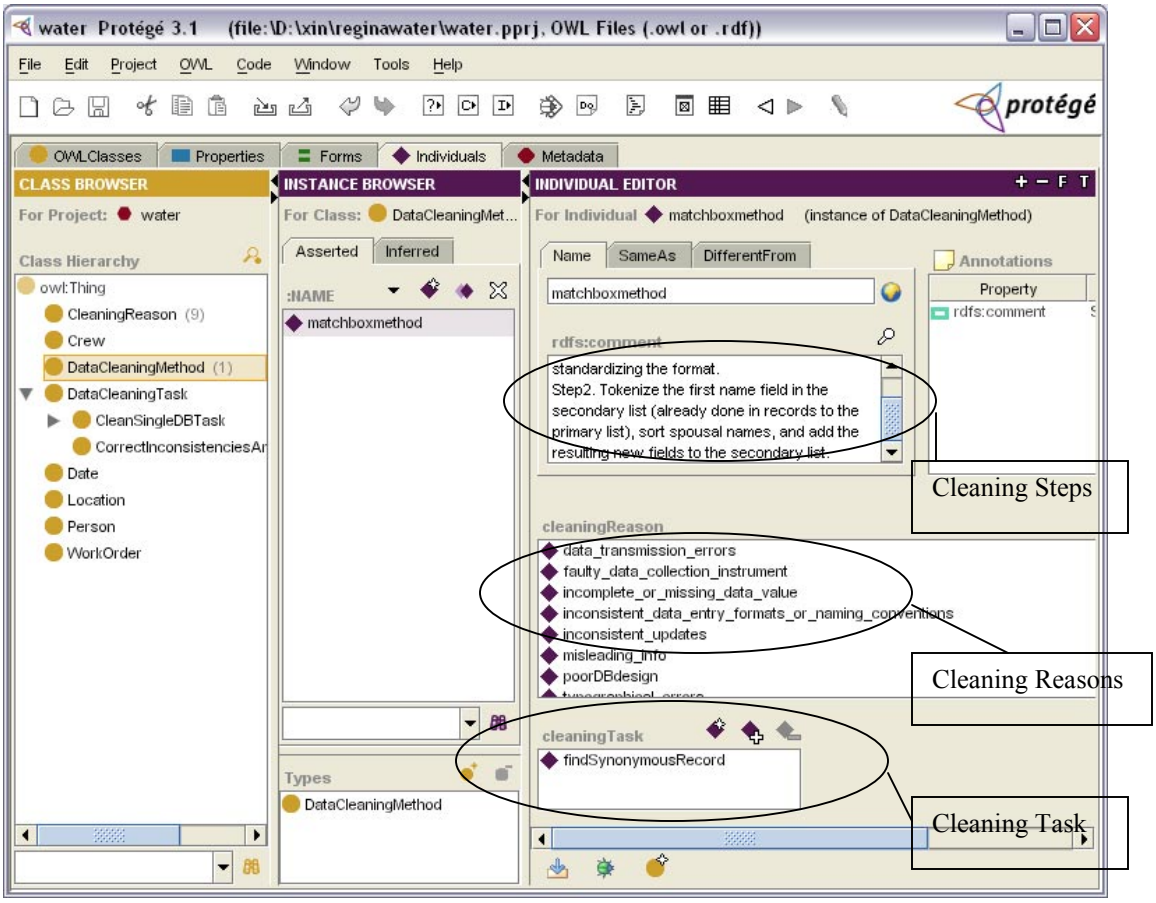

Figure 2. MatchBox method is an instance of DataCleaningMethod class

In the ontology, the method instance also provides the steps that the data cleaning task needs to follow. At present, we put the procedure as comments. For example, consider the Matchbox algorithm for the synonymous record problem. Given a primary list containing known, valid descriptions of contact information and a secondary list of possibly new, possibly invalid descriptions of contact information, the Matchbox algorithm consists of the following steps.

1. Repair all records in the secondary list by removing any typographical errors and standardizing the format. Repair the street addresses by checking them against external sources listing valid street addresses, city names, province abbreviations, postal codes, etc. For example, “PQ” and “QC” are often used to designate the province of Quebec, but Canada Post’s official code is “QC”. Similarly, “NL”, the province code for Newfoundland and Labrador, should replace instances of “NF” [3]. This repair process is called BestRepair.
2. Tokenize the first name field in the secondary list (already done in records to the primary list), sort spousal names, and add the resulting new fields to the secondary list. The new fields that are added to the list are canonical first names (such as “Robert” for “Bob”), the initial of the first name (First Initial) and the Soundex version of the last name (Soundex Last).

3. For each conditional rule specifying a condition under which a match should be made: Remove any record from the secondary list that matches a record in the primary list according to the current conditional rule.
4. For each remaining record in the secondary list: Remove any record that the user can match manually to a record in the primary list.
5. Place all matches between records in the primary and secondary lists found in step 3 and 4, in the match list, for later processing by a separate merging process, possibly including user intervention.
6. Place the unmatched records from the secondary list that are in valid format in the novel list, for later combination with the primary list.

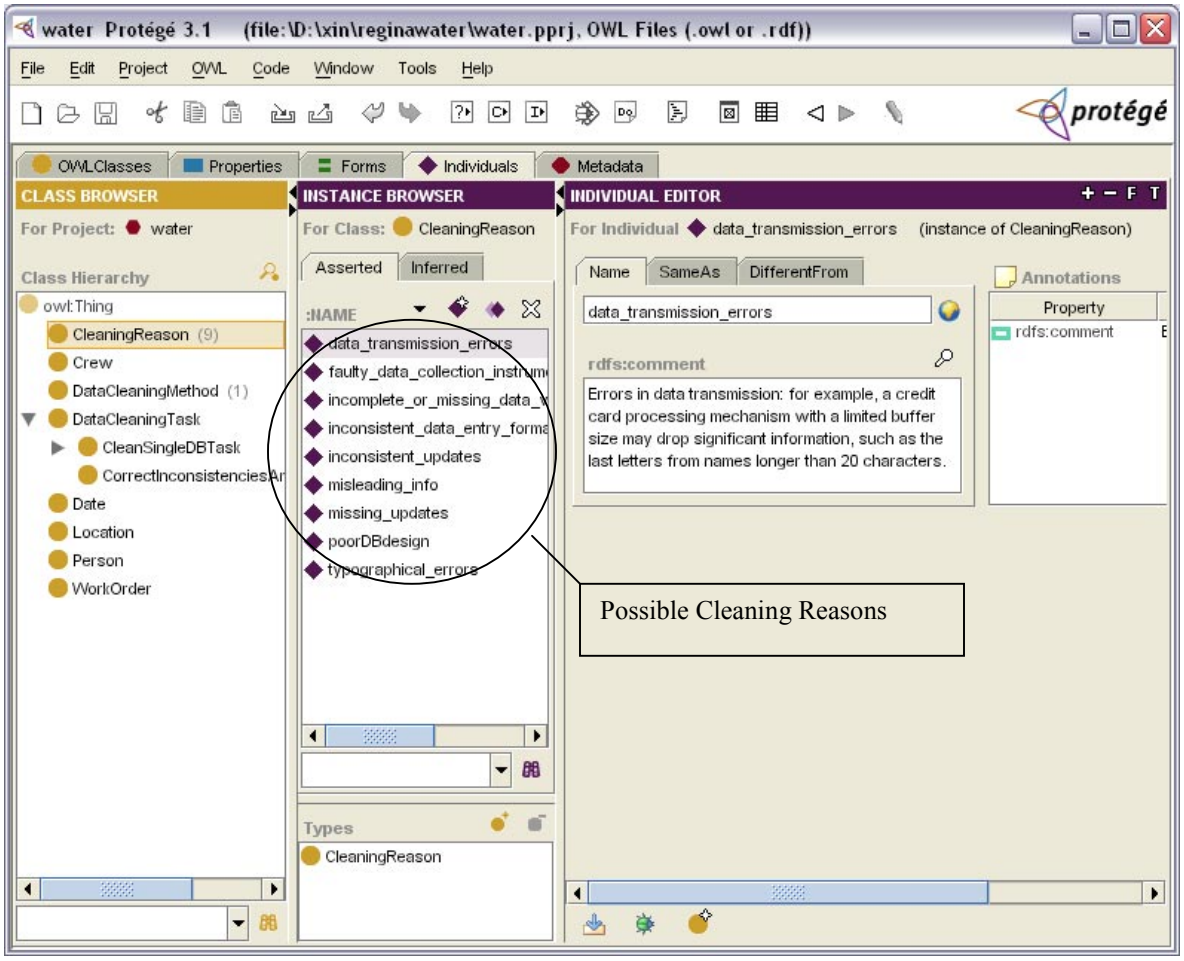

Figure 3. CleaningReason class lists reasons for cleaning

3) Another important top class in the data cleaning ontology is the `CleaningReason` class that lists the reasons for cleaning. After the cleaning, we can explain the cleaning results using the ontology. For example, as shown in Figure 3, after we apply MatchBox to address the synonymous record problem, the cleaning result might be given in terms of the following nine possible explanations [2][7][14].

1. Typographical errors in data entry and data recording: for example, the first name may be recorded as “Hohn” instead of “John”.
2. Inconsistent data entry formats or naming conventions: for example, the apartment number may be recorded as a separate field in one list, but be combined with address line 1 in another list.
3. Inconsistent updates: for example, two lists may both include product codes to categorize items, but the set of product codes may have been changed, resulting in two different codes for same product in the list.
4. Poor database design: for example, a field may not have been provided to record middle names.
5. Faulty data collection instrument: for example, due to poor software design the first name may always be stored as the last name and the last name as the first name.
6. Errors in data transmission: for example, a credit card processing mechanism with a limited buffer size may drop significant information, such as the last letters from names longer than 20 characters.
7. Missing updates: for example, a woman who changed her name when she got married may not have made an explicit request that her last name be updated.
8. Misleading information deliberately provided by a customer: for example, a person may have given his nickname instead of his legal first name in an effort to be treated as two different individuals.
9. An incomplete or missing data value: for example, a customer may not have provided his middle name.

Besides the general classes of data cleaning tasks and methods, a domain ontology also provides domain constraints relevant to data cleaning.

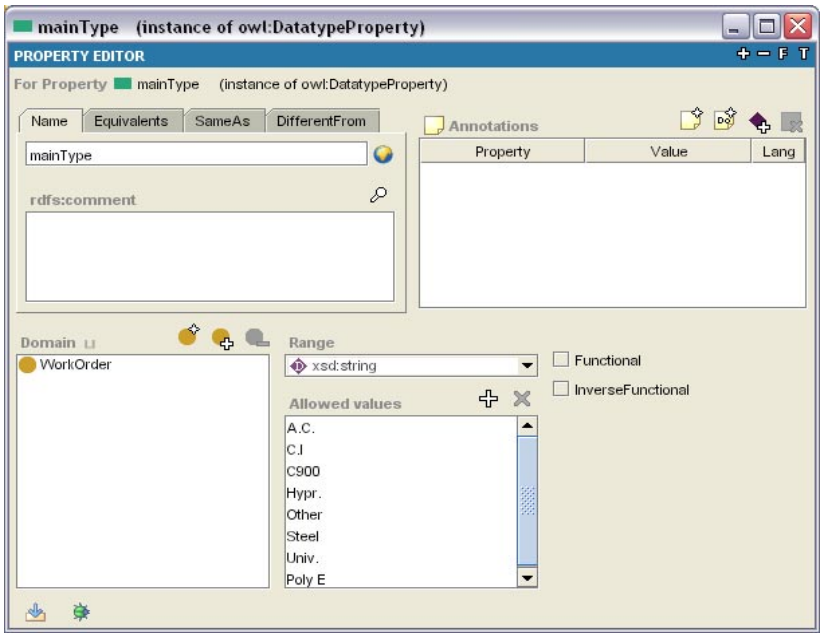

Figure 4. The mainType attribute in WorkOrder class

If our data contain errors such as misspellings, missing values and misplace values, we can use ontology to check the domain constraints on the attributes. For example, as shown in Figure 4, the mainType property in WaterOrder class can only be assigned eight possible values: *A.C.*, *C.I.*, *C900*, *Hypr.*, *Steel*, *Univ.*, *Poly E.*, and *Other*. The ontology lists all those values, which could use to check invalid values for the mainType attribute. For another example, if the value of *City* is set to *SK*, the ontology will suggest a cleaning method that can attempt to find a correct city name, since *SK* is not a valid city name.

A domain ontology can also help us to check for some other semantic errors. For example, given City = "Regina", Postal Code = "S4S 3A2"; Postal Code refers to a Moose Jaw address. This case violates the attribute dependency in the ontology. As another example, if ProblemType = "Corrosion", PipeType = "PVC", a referential integrity check will show corrosion is not possible on a PVC water main.

5. Conclusions

In this paper, we presented OntoClean, an ontology-based data-cleaning framework. Our approach to data cleaning requires a set of ontologies describing the domains represented by the classes and their attributes. Using the ontology-based approach, we are able to clean data at the knowledge level instead of data level.

Future work includes investigating available data cleaning methods and connecting them with appropriate data cleaning tasks. Determining how to represent the constraints on specific attributes for a specific domain is another topic that requires further research.

References

[1] Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web. Scientific American 284(5) (2001) 34-43

[2] Bither, Y. *Cleaning and Matching of Contact Lists Including Nicknames and Spouses*, M.Sc. Thesis, Department of Computer Science, University of Regina, September 2005.

[3] Galhardas, H., Florescu, D., Shasha, D., and Simon, E., AJAX: An Extensible Data Cleaning Tool, *Proc. 2000 ACM SIGMOD Conf. Management of Data* (*SIGMOD '00*), Dallas, 2000, page 590.

[4] Galhardas, H., Florescu, D., Shasha, D., Simon, E., and Saita, C.A. Declarative Data Cleaning: Language, Model, and Algorithms. In *Proceedings of the 27th International Conference on Very Large Databases* (*VLDB 2001*), pages 371-380, Rome, Italy.

[5] Gruber, T. R.: A translation approach to portable ontologies. Knowledge Acquisition, 5(2) (1993) 199-220.

[6] Hair, J.F., Anderson, R.E., Tatham, R.L., and Black, W.C., *Multivariable Data Analysis*, Fifth Edition, Prentice Hall, 1998.

[7] Han, J., and Kamber, M. *Data Mining: Concepts and Techniques.* Morgan Kaufmaan, 2001.

[8] Hernandez, M.A., and Stolfo, S.J. Real-world Data is Dirty: Data Cleansing and the Merge/Purge Problem. *Data Mining and Knowledge Discovery*, 2(1): 9-37, 1998.
[9] http://www.artsci.wustl.edu/~philos/MindDict/ontology.html, accessed June 10, 2005.
[10] http://protege.stanford.edu/index.html
[11] http://protege.stanford.edu/plugins/owl/index.html
[12] http://www.tulane.edu/~panda2/Analysis2/datclean/dataclean.htm. Accessed on May 13, 2005.
[13] http://www.w3.org/2001/sw/WebOnt/
[14] Lee, M.L., Lu, H., Ling, T. W., and Ko, Y.T. Cleansing Data for Mining and Warehousing. In *Proceedings of the 10th International Conference on Database and Expert Systems Applications* (*DEXA'99*), Florence, Italy, pages 751-760, Aug 1999.
[15] Low, W.L., Lee, M.L., and T.W. Ling, T.W. A Knowledge-Based Approach for Duplicate Elimination in Data Cleaning, *Information Systems*, 26(8):585-606, Dec. 2001.
[16] Raman, V., and Hellerstein, J.M., Potter's Wheel: An Interactive Data Cleaning System, In *Proceedings of the 27th International Conference on Very Large Databases* (*VLDB 2001*), pages 381-390, Rome, Italy.
[17] Sund, R.: Utilisation of administrative registers using scientific knowledge discovery. Intelligent Data Analysis, 7(6) (2003) 501-519
[18] Wang, X., and Hamilton, H.J., Towards an Ontology-Based Spatial Clustering Framework, In *Proceedings of 18th Conference of the Canadian Society for Computational Studies of Intelligence* (*AI 2005*), Victoria, BC, Canada, May 2005, 205-216.
[19] Zhang, S., Yang, Q., and Zhang, C.Q. (Eds.), *Proceedings of the First International Workshop on Data Cleaning and Preprocessing*, Maebashi City, Japan, December, 2002.