Send Orders for Reprints to reprints@benthamscience.ae

The Open Automation and Control Systems Journal, 2015, *7*, 2039-2043

Research and Realization of the Extensible Data Cleaning Framework EDCF

Xu Chengyu* and Gao Wei

State Grid Shanxi Electric Power Corporation, Shanxi, Taiyuan 030001, China

Abstract: This paper proposes the idea of establishing an extensible data cleaning framework which is based on the key technology of data cleaning, and the framework includes open rules library and algorithms library. This paper provides the descriptions of model principle and working process of the extensible data cleaning framework, and the validity of the framework is verified by experiment. When the data are being cleaned, all the errors in the data source can be cleaned according to the specific business by the predefined rules of the cleaning and choosing the appropriate algorithm. The last stage of the realization initially completes the basic functions of data cleaning module in the framework, and the framework which has good efficiency and operation effect is verified by the experiment.

Keywords: Data Cleaning, Clustering, Outlier, Approximately Duplicated Records, The Extensible Data Cleaning Framework.

1. INTRODUCTION

Due to the complexity of the data cleaning, for different data sources, different data types, data quantity and specific business are required while cleaning data. No matter how effective measures a data cleaning algorithm adopts, it can't give total good cleaning performance on all aspects, and it can't just rely on one or several algorithms to solve all kinds of data cleaning problems with generally good performance. In addition, the detection and elimination of some data quality issues are rather complex, or associated with specific business. Thus, in-depth inspection and analysis to deal with these types of errors should be conducted, even though not all the errors contained in data can be detected and eliminated. Because each method can detect different error types and different ranges, in order to check out the error as many as possible, a variety of methods should be adopted for error detection at the same time.

From the above analysis we can see that it is necessary to provide an expansible software which contains a series of data cleaning algorithm and auxiliary algorithm (used for data preprocessing and visualization) and can use specific business knowledge. Thus it can provide different cleaning methods and algorithm supports for the data cleaning with different backgrounds. The author absorbs the thought of literature, he call it an Extensible Data Cleaning Framework (namely EDCF). Realizing a data cleaning software framework based on the process which contains rich data cleaning methods library, data cleaning algorithms library and extensible cleaning rules, is helpful for us to utilize the prior knowledge rules to take advantage of their merits and avoid demerits among different methods and algorithms. Therefore, it is convenient for the user to select suitable problem cleaning method from rich software tools so as to improve the cleaning effect of data cleaning algorithm in different applications [1-3].

Therefore, based on the demerits in the data cleaning methods put forward by former people, this paper puts forward an extensible data cleaning framework. This framework has open rules library and algorithms library; besides, it provides a large amount of data cleaning algorithm as well as other data standardization methods. At the moment data source is going through cleaning, according to specific business, with the predefined cleaning rules this framework can choose the appropriate algorithm to clean all the errors in the data source. Therefore, this framework has strong versatility and adaptability, and it also greatly improves the comprehensive effect of the data cleaning.

2. THE PRINCIPLE OF THE EXTENSIBLE DATA CLEANING FRAMEWORK

2.1. The Function Modules and Cleaning Methods of EDCF

Data cleaning framework includes the common data cleaning function module and the corresponding main cleaning algorithms. EDCF's main cleaning function modules and its related cleaning algorithms are as follows: data standardization modules

Before we complete a data cleansing function, in order to unify the format of data in data source, we need to standardize the records in the data source. Data standardization operation plays an important role in improving the detection efficiency of subsequent error data and approximately duplicated data, reducing the complexity of time and space. So, the author makes some data standardization methods as a preprocessing method of data cleaning in EDCF. When the

*Address correspondence to these authors at the State Grid Shanxi Electric Power Corporation, Shanxi, Taiyuan 030001;
E-mail: Hunter2011@foxmail.com

1874-4443/15

data need to be cleaned, firstly, we should complete the data standardization [4], and we can directly invoke this function. For data standardization method, error data cleaning modules (outlier data cleaning modules).

About the incomplete data and error data cleaning problems, at present, we mainly use the following two methods, namely the outlier detection and business rules. To test cleaning module of approximately duplicated records about how to solve the cleaning problem of approximately duplicated records, it can be summarized as follows. As far as approximately duplicated records are concerned, there are two ways which can be used to detect. One is to use sorting comparison method to detect. This method sorts the recorded data in a table by given the order first, and then by comparing the similarity degree of the adjacent records to detect approximately duplicated records. The other is to use the clustering analysis. According to the principle that approximate records will be clustered into the same cluster, we can detect approximately duplicated records.

2.2. Cleaning Process

The specific cleaning process of extensible data cleaning is as follows:

The first stage: in the data analysis phase we analyze the data source needed to clean, define the rules of data cleaning, and select the appropriate cleaning algorithm to make it be better adapted to the data source needed to clean.

The second stage: data standardization phase we put the data in data source which need to clean into the software framework by JDBC interface transferring. We use the corresponding rules in the rules library to standardize the data source, such as to standardize data record format, etc. And according to the predefined rules, we transfer the corresponding data record field in the same format.

The third stage: data cleaning phase according to the analysis of the data source, we perform data cleaning operations step by step, and the general cleaning process is as follows. First of all, we clean error data, and then we clean the approximately duplicated records. Here the sorting comparison of the approximately duplicated records processing module can also be regarded as a kind of simple clustering operation.

The fourth stage: treatment phase of cleaning result after the data cleaning operations, the data cleaning results will be shown in the system window. According to the result of cleaning and the warning information, manual cleaning doesn't conform to the data of predefined rules, and its algorithm cannot meet the data requirements, thus data cleaning system is used. In addition, by going through the data cleaning log, the data cleaning effect can be seen [5]. Thus, the correctness of the data cleaning can be verified, and also the wrong data of the cleaning can be modified.

In the working process of the above four phases, the user can choose the following contents:

(1) Rules used to carry out data standardization

(2) Record field to make comparison

(3) Clustering algorithm

(4) Fuzzy rules used in fuzzy calculation system

(5) Merge strategy

In each step of the above mentioned process, the user can, from the existing "methods library" (such as existing algorithm, rules, etc.), select a corresponding method or add his own way to the "methods library." Firstly, the user selects a number of data standardization rules to standardize the data which enter data cleaning phase. Secondly, for the standardized data records, the user can choose the attributes which are more important in his opinion. The relative clustering operations will go on based on the user's choice of the properties represented by the records. Thus, the efficiency of data cleaning can be improved and the complexity of the algorithm can be reduced. Thirdly, the user selects an algorithm to cluster the record which has already been sifted, and the clustered records may be the approximately repeated records [6]. While the records which are not in the clustered records and the records which are in the clustered records but far away from the clustering center may likely to be the outlier records. Fourthly, according to the result of clustered records and the preset threshold value, based on the fuzzy rules we determine whether the records are approximately duplicated records or outlier records. Lastly, approximately duplicated records are grouped together. Because there are different merger strategies, the user must choose one merger strategy for many approximately duplicated records and select a representative to record.

2.3. EDCF Rule Library and Algorithms Library

2.3.1. Rule Library

Rules library and algorithms library are the core of the extensible data cleaning software framework. Among them, the rules library is used to store the rules about data cleaning, and it mainly includes:

(1) business rules

Business rules refer to a certain scale, a set of valid values, or a certain mode of one business, such as address or date. Business rules can help to deal with the exceptions in testing data, such as the value which violates the attribute dependence, or the value beyond the scope, etc.

(2) duplicated records recognition rules

Repeat identification rules are used to specify the condition that there are two approximately duplicated records; For example, the threshold value of field similarity-$\delta 1$, the threshold value of field edit distance-k, the threshold value of record' similarity degree-$\delta 2$, the threshold value of high-dimensional space similarity-δ, radius-ε, point set number-Minpts, etc.

(3) error records identification rules

Error records identification rules are used as one of the pre-conditions of specifying a piece of wrongly recorded data, such as the following parameters defined in outlier detection, α, β, γ, δ etc.

(4) merger/clear rules

Table 1. The Table Structure of the Rule Library Data Tables.

Table Name : RULE_TABLE				
Text Fields	**Type**	**Length**	**Description**	**Key**
Ruleno	CHAR	10	Rule Number	PK
Rulename	CHAR	10	Rule Name	
Ruledata	CHAR	10	Rule Value	
Ruleexplain	VARCHAR	50	Rule Description	

Merger/clear rules are used to define how to deal with a set of approximately duplicated records. There are two kinds of processing methods used to deal with a series of records which are detected to be approximately duplicated. One is to treat a record as a correct one, while other records as duplicated records which contain false information. Another is to make each duplicated record as a part of data source, whose purpose is to merge the duplicated records, and produce a piece of new record with a more complete information. The processing of approximately duplicated records is accomplished by the user according to the merger/clear rules predefined in rule library in specific business [7].

(5) warning rules

Warning rules are used to specify the processing rules and the corresponding message for some special event.

In the data cleaning process of data source, according to specific business, the warning rules define the corresponding rules in rules library, or modify the existed rules. Thus we make the software framework be suitable for different data sources. These rules have strong versatility and adaptability.

For the realization of the rules library, the author divides this process into the following two ways:

Method 1: It is accomplished through the rule language

Rule language generally adopts the IF-THEN rules, and this approach is mainly used in cleaning rules, such as business rules, warning rules.

Method 2: It is accomplished through the establishment of a table in the data library.

This method is mainly used in cleaning rules, such as repeated recognition rules, errors recognition rules etc. The table structure of data table is shown in Table **1**.

Data cleaning framework provides the definition interface of data cleaning rule to define or modify data cleaning rule in the rule library according to the specific business analysis in the process of data cleaning.

2.3.2. Algorithms Library

The algorithms library is the algorithm needed to be used to store data cleaning. After the realization of the program, various data cleaning algorithms are stored in the algorithms library in the form of class. When we do the data cleaning, we can invoke the corresponding suitable algorithms according to different situations. After the programmatic realization, the required new algorithm is added into the algorithms library [8]. The general form of the algorithm is as follows.

```
PUBLIC CLASS ARITHMETIC_Name
{
Data type Data name1 ;
Data type Data name2 ;
Return Data type Method1 ;
Return Data type Method2 ;
}
```

In them, ARITHMETIC_Name represents the name of the algorithm and PUBLIC shows the public class, that is to say, the algorithm ARITHMETIC_Nam can be invoked anywhere and it is not limited by the scope of program module.

The author uses a rectangular which is divided into three regions to represent the object class of the algorithm. In them, three regions show respectively the class name of algorithm, the attribute of algorithm and operation of algorithm. Repeating the cleaning of the data source for many times by selecting the appropriate cleaning method can improve the comprehensive effect of data cleaning. In addition, some new data cleaning algorithms can be continuously expanded in the algorithm library to be used for data cleaning.

3. THE REALIZATION OF EDCF

Like the development of other application systems, the system modeling is very important in the process of developing EDCF. In the development of EDCF, we use UML (Unified Modeling Language) and Rational Rose to do system Modeling so as to ensure the scalability of the EDCF. The understanding and the description of the issues about the analysis phase of the traditional software engineering system don't take the inherent things in the problem domain as the basic unit to analyze but take the function, the data and the data flow in the global scope as the center to analyze. It is easy to cause the deviation in the understanding and analysis results. The emergence of UML provides a new method to avoid and reduce this kind of deviation. UML is a kind of method which uses the unified and standardized tabs and definitions to realize the design and analysis of the object-oriented software. It uses a set of mature modeling technology to be suitable for all the stages of software development. It integrates object modeling technique, object oriented software engineering and other design and analysis methods. It

is a standard modeling language for the object oriented technology. UML applies to all aspects of the software analysis and design. It is composed of view, graph, the model element, the general mechanism and other several parts. Views are used to represent various aspects of the modeled system, and consists of multiple graphs. The graphs are formed by all kinds of pictures. They are used to describe the content of a view. UML language defines nine different kinds of graphs. We combine them together organically so that we can describe the system well. The model elements represent the concepts such as the object oriented class, object, message, relationship and so on. They are the most basic and the most commonly used concepts which constitute the graphs. The general mechanism is used to indicate additional information, such as annotation, semantic annotation of model elements. UML also provides an extensible mechanism. In order to rapidly develop EDCF with high quality, the use of visual modeling tool is very important, while Rational Rose is the best modeling tool based on the UML development application system. Rational Rose is a powerful and visualized modeling tool with the support of UML, which is launched by Rational Rose company. It uses the use-case, logic, components and deployment views to support the analysis and design of object oriented software. Establishing the corresponding UML graphics in different views can reflect the different characteristics of the system [9, 10]. The forward, reverse engineering function provided by Rational Rose can convert between the system UML design model and system language code. After we use Rational Rose to do UML modeling for a system, we can get the corresponding framework codes of object-oriented language by forward engineering. This development method accelerates the development speed of the system and guarantee the quality of the system. Therefore, this paper uses Rational Rose to do EDCF modeling based on the UML. The whole modeling process is described as follows:

First we carry on the system analysis in this paper and design use-case model of each module in EDCF. According to the use-case model of each module, we can find out the use-cases needed to realize so as to design "Realize model". According to the "Realize model" and the basic process of each module, we can design Sequence diagram and collaboration diagram of each Realize model, and then we can abstract the class diagram. And we can integrate the same and repeated class. And the last one is the physical design, so we can complete the modeling work of EDCF.

When we do the system modeling of the EDCF, we should pay attention to the following problems.

1. In addition to using Sequence diagram and collaboration diagram, we can also use the state diagram, activity diagram, etc. But the built model can accurately reflect the system [11].

2. We should ensure the accuracy of the UML model, because the wrong model leads to form the error code.

3. When the analysis phase makes the function of the system be refined to each component, we should carefully consider. The inappropriate functional division will lower the quality of the program and increase the difficulty of system maintenance.

4. We should establish the UML model of the system as detailed as possible. Thus, after the forward engineering, it will generate more detailed code framework.

Based on the above analysis, we can see that when we develop the EDCF, we can adopt the UML as a modeling language. As the analysis and design tool, Rational Rose can finish the complete transition from system analysis to system design and coding. This kind of development method provides a high starting point for the development of EDCF to accelerate the development of the EDCF and improve the quality of the system, and lay a good foundation for the system expansion and maintenance in the future so as to develop high-quality, reusable, Extensible EDCF.

After we use Rational Rose to complete the system modeling of the EDCF, we can get the corresponding framework codes by the forward engineering function of Rational Rose. On this basis, we can adopt Microsoft visual Studio 2005.NET as the development platform to be able to complete the development work of the EDCF.

4. THE EFFECT EVALUATION OF THE DATA CLEANING FRAMEWORK

In order to further verify the effectiveness of the framework, we use the commercial data in the foreign trade processing system from Shandong World Trade Center as the simulation to be the input data set. It mainly contains 52681 data records about commodity name. And the length of the name is in the range of 2-21. And the average length is 8. We use database generator in these data to generate respectively 35% duplicate records and 20% error data randomly. The scale of the data increases progressively according to the sequence of 10000, 20000, 30000, 40000, 50000. The value of σ is 0.75. The experiment is carried out in the environment of PC CPU Pentium III 800 windows XP RAM 512M.

For the detection experimental part of error data records, we adopt three different methods to compare the experimental results which we get. For the detection experiment part of approximately duplicated records, we adopt the experimental results which we get by using the second method to compare. The first one is to take a single merge/purge method. And it is a comparatively representative and better method among the traditional methods. It makes the entire database records be given order according to the dictionary then be clustered after that. It is known as the merge/purge method here. The second one is to adopt the data cleaning method of extensible framework in this paper to detect approximately duplicated records. The experimental results are shown in [12-14].

（1）The data cleaning system can realize the data cleaning tasks of data source well, and improve the quality of the data source well, and provide the quality guarantee for establishing an efficient decision support system.

（2）The introduction of data preprocessing module in the data cleaning system is feasible and necessary, therefore, preprocessing operations can significantly improve the efficiency and effectiveness of data cleaning.

CONCLUSION

Data cleaning is the important work that needs to be performed before the enterprise builds the warehouse. But due to the complexity of the data cleaning, data cleaning has great difficulty. Based on the research of the previous paper, this paper puts forward an extensible data cleaning framework which has open rules library and algorithms library. The rules library is used to store cleaning rules. And algorithms library is used to store the cleaning algorithms. The algorithms library which can be extensible contains many kinds of algorithms. The software framework is applied to different data sources by defining the cleaning rules in the rules library and choosing appropriate cleaning algorithms from algorithms library so as to make it have strong versatility and adaptability. By the cleaning of various algorithms, we can improve the comprehensive effect of data cleaning. Finally, the key technology of the software framework is realized. And we make it be applied to the cleaning of simulated business data set. The experiment verifies the effectiveness and the feasibility of the framework. With the progress of the research, the author will further improve the software framework, and make the cleaning function be kept increasing.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCE

[1] V. Barnett, and T. Lewis, "Outliers in Statistical Data," New York : John W iley and Sons,1994.

[2] E. M. Knorr, and R. T. Ng, "Algorithm s for Mining Distance-based Outliers in Large Datasets," *New York: Proc. of Int. Conf. Very Large Databases* (VLDB'98), pp. 392-403, 1998.

[3] E. Knorr, and R.A. Ng, "Unified Approach for Mining Outliers: Properties and Computation," *Newport Beach: Proc. of Int. Conf. Knowledge* Discovery and Data Mining(KDD'97), pp. 219-222, 1997.

[4] S. Jiang, Q. Li, K. Li, and H. Wang, "GLOF: A New Approach for Mining Local Outlier," In: *International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 157-162, 2003.

[5] S. D. Bay, M. Schwabacher, "Mining Distance based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule," Washington DC: SIGKDD' 03, pp. 29-38, 2003.

[6] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets," In: *Proceedings of the ACM SIGMOD Conference*, vol. 29, pp. 427-438, 2000.

[7] M. M. Breunig, H.P. Kriegel, R. T. Ng, and S. Jörg, "Optics of : Identifying Density based Local Outliers," Zytkow J M, Rauch. "Proc. of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases," *Lecture Notes in Computer Science*, vol. 1704, pp. 262-270, 1999.

[8] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast Outlier Detection Using the Local Correlation Integral," In; *The 19th International Conference on Data Engineering*, vol. 315, pp. 315-326, 2003.

[9] J. W. Tukey, "Exploratory Data Analysis," MA: Addison-Wesley and Sons, Inc., 1994.

[10] H. Li, Y. Liu, and Y. Li, "Application of fuzzy equivalence theory in data cleaning," *Journal of Southeast University (English Edition)*, vol. 20, no. 4, pp. 454-457, 2004.

[11] T. F. Smith, and M. S. Waterman, "Identification of common molecular subsequences," *J Molecular Biology*, vol. 147, pp. 195-197, 1981.

[12] R. A. Agner, and M. J. Fischer, "The string-to-string correction problem," *JACM*, vol. 21, no. 1, pp. 168-173, 1974.

[13] M. A. Hernandez, and S. J. Stolfo, "The Merge/Purge Problem for Large Databases," *SIGMOD*, vol. 24, pp. 127-138, 1995.

[14] A. H. Mauricio, and S. J. Stolfo, "Real-World Data is Dirty: Data cleansing and The Merge/Purge Problems," *Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 9-37, 1998.

Received: June 16, 2015 Revised: August 10, 2015 Accepted: September 19, 2015