

ENHANCING DATA PREPARATION PROCESSES USING TRIGGERS FOR ACTIVE DATAWAREHOUSING

Kanana Ezekiel

Department of Computing, Communication Technology
and Mathematics, London Metropolitan University,
London, UK
k.ezekiel@londonmet.ac.uk

Farhi Marir

Department of Computing, Communication Technology
and Mathematics, London Metropolitan University,
London, UK
f.marir@londonmet.ac.uk

Abstract: Data preparation is a significant pre-processing task to prepare data for mining. The data mining process cannot succeed without a serious effort to prepare data. Very often mistakes are found in data, thus making the analysis process more difficult. Without the data preparation phase, we will have no idea whether the data quality can support analysis queries. Several techniques exist for data preparation in data warehousing. However, one of the problems of existing approaches is their limited support for data preparation for active and changing environments such as Active Data Warehouses. Their focus is on static data preparation approaches. This paper addresses this limitation and a trigger mechanism designed to manage changes in a dynamic environment is utilized. The specification language of a trigger supports active and dynamic capabilities that enable users to automatically filter or select and cleanse data at runtime. In addition the focal point of this work is not only on syntactic but also semantic data preparation approach.

Index Terms— Triggers, Data preparation, Data cleaning, Data Mining, Active Data Warehouse, Active Database System, Event Condition Action (ECA) Rules

I. INTRODUCTION

Over the years, people viewed their data warehouses as static as the data did not change very often.

Then it evolved, as the static data sets could not give the most current and recent changes necessary. The data warehouse environment was set up to give static snapshots of data at some point in time, perhaps from as recently as the last week. However last week's data or even last night's data is often not sufficient to react to current situations. Things change rapidly in today's electronic-business economy and the company with the best set of integrated, current data is the one that will survive. Organizations today are spending a lot of money on acquiring a comprehensive integrated set of

accurate data from their data warehouses to enhance performance and outstrip the competition.

Accommodating a customer within minutes of an event represents the behavior of an active database system. Active Data Warehouse (ADW) applies the idea of Event-Condition-Action rules (ECA rules) from active database systems to implement active behavior [1]. In most active database systems, triggers also known as Event-Condition-Action rules are used to monitor changes and enforce integrity constraints. For active data warehouse environments, the ECA rules are required for checking the data and automating routine analysis decisions. As data warehouse environments adapt to new active advancements, a significant need exists for new techniques with abilities to automatically help users to prepare data for mining in active and evolving environments. Data preparation is a significant pre-processing task that is carried out as part of data mining and before analysis phase. The objective of data preparation phase is to cleanse and transform data into a format suitable for the application and analysis phase. The analysis process cannot succeed without a serious effort to prepare data, as very often mistakes are found in the data collected, which sometimes presented in an unstructured form. Furthermore, data needed may not be readily available due to rapid changes, thus making the analysis process more difficult. Without the data preparation phase, we will have no idea if the data quality can support analysis queries. Several techniques exist for data preparation in data warehousing (Section II). However, one of the problems of existing approaches is their limited support for data preparation in active and evolving environments; their focus is on static snapshots of data. Data preparation tends to be a one-off task before data analysis. The difficulty is that, the way data is cleansed depends on the intended use expressed as a business purpose for analysis. If the business purpose changes through time then the data preparation process also needs to adapt to change. For example, hospital records may contain sufficient

information to determine the country of residence of a cancer patient. However, it may not contain a full address and so be “dirty” for posting purposes. If the data preparation process is not well defined/designed, there is always the possibility of inappropriate cleaning for a particular purpose. Furthermore, data preparation is considered as one of the most time consuming activities in a passive data warehousing. If we are not careful the problem could be doubled in an active data warehousing. Businesses could lose money while waiting for data to be cleaned.

This paper addresses this limitation and a trigger mechanism designed for data preparation in a dynamic environment is proposed. The specification language of a trigger mechanism supports active and dynamic capabilities to enable users to filter relevant information and cleanse data as you load data into a data warehouse. The idea of incorporate triggers with dynamic capabilities using the Dynamic Object Model concept (DOM) of Object Oriented modeling techniques to support changes in active databases first appeared in [2]. A trigger object holds information about dynamic objects that are made up of Event, Condition and Action components. With the ability to store triggers and their components as dynamic objects we can devise metadata and provide flexible concepts for in advance definition and manipulation. So a trigger implemented to perform certain data preparation activities can be easily added, deleted or changed since the underlying concepts are stored in a database. Here the focus is on efficient mechanisms to manage triggers that perform data preparation therefore managing semantics of data preparation processes.

The rest of the paper is organised as follows: Section II looks at the background and related works. In section III, we present our motivation. Section IV presents the outline of our proposed approach and finally future work and main conclusions are given in section V.

II. BACKGROUND AND RELATED WORK

Data preparation process involves several subtasks: information filtering, integration, transformation, cleaning as well as reduction of data. Information filtering or data selection identifies the relevant data for the analysis phase. Data integration combines data from different sources to form new values. Data transformation becomes necessary in situations such as migration and integration of data from legacy or multiple data sources. Data transformation supports any changes in data structure, syntax or content of data. Data cleaning also called data cleansing deals with data quality problems by selecting clean subset of data, filling in missing values, resolving inconsistencies and eliminating errors. Data quality problems are present due to misspellings during data entry, duplicate

information collected from multiple data sources, inconsistencies, unstructured data (data format) collected from multiple data sources, etc. Finally data reduction deals with compression of data in order to reduce the analysis effort.

In recent years there have been a lot of efforts to support data preparation tasks for data warehousing and data mining. Although not all data preparation activities can be solved by using automated tools, a variety of software tools that address different kinds of data preparation tasks ranging from general to specific-purposes tools are on the market [3-6]. These tools include database design tools and Extraction-Transformation-Loading (ETL) tools. Also available on the market from companies such as IBM, Ascential, Trillium and others are so called data quality and auditing tools that help analysts to detect and eliminate inconsistencies. However, all these tools are not designed for dynamic environments. Active data warehouse evolves as data change rapidly. Those changes need to be maintained and tracked through the lifespan of the system. The problem with data is that its quality quickly deteriorates over time. Experts say in the world of continuous competition, data quickly becomes obsolete. Therefore the demand to support dynamic data preparation models is inevitable. Our approach uses triggers with dynamic capabilities to support data preparation tasks in active or dynamic environments. Using triggers with dynamic capabilities will allow user to quickly create and modify rules to speed up the filtering and cleansing tasks. It will also increase usability.

Besides the software tools mentioned above, there has been a lot of work done by various researchers. Several approaches addressing data cleaning problems were proposed in [7 - 14]. There are few solutions to data cleaning problems but a common approach is to write a script in language such as C, ProC, Java, Perl, etc, to implement the whole data cleaning process. Unfortunately mentioned solution is not flexible enough to support the dynamic nature of data cleaning processes discussed in this paper. Also this solution does not take the advantages of features offered by existing databases such as triggers, hence increasing data preparation costs.

Another approach for selecting and detecting duplicates as part of data cleaning process is described in [11]. In [11] the Duplicate Elimination Sorted Neighborhood Method (DE-SNM) is used for detecting data with exact duplicate keys first. The effective of this method depends on the key selected to sort records. The main drawback to the DE-SNM method is its static nature and usability limitation. Finding the suitable key for putting together similar records may be quite hard. Also some work has been done in extending GROUP BY operator and user-defined aggregates for duplicates elimination. The GROUP BY operator supports

grouping on uninformed expression. The user-defined aggregate merges the final grouped records into one record. Moreover, there are other similar attempts proposed for processing large data sets [15] but we are not aware of any method that deals with active or dynamic warehouse environments. Furthermore, there are some works on SQL query language for data preparation and analysis tasks. Given a database that adheres to a relational model, data integrity such as entity and referential integrity can be used as a simple data cleaning process. Relational data integrity can be implemented using SQL queries. Through standard SQL basic features, one can filter, uncover errors and transform data. Many recognized database management systems such as Oracle, Sybase, DB2, Informix, Access, etc, support this type of data cleansing. The work presented in [16] presents SQL/MX a query language offering features like transposing and sampling for data reduction. Our approach is partially inspired by query implementations but combines query features with active features.

There are few data preparation proposals in the World Wide Web and data-mining environment (also referred to as Web mining) [17-23]. The Web mining environment faces similar problems. The development of Internet services often requires integration of large amounts of data that change rapidly from client with unpredictable needs. Web data is collected in various ways so there is a need to preprocess data to make it easier to mine knowledge. Existing approaches are inefficient for rapidly changing data. The focus is on the syntactic aspect of correcting data while our approach the focus is on the semantic aspect using triggers to correct data. By manipulating and modifying triggers, we are correcting the meaning of data making it ready for analysis in short time avoiding processing delays. The processing delays can be significant if not taken care, making it difficult to achieve acceptable responses. The effort needed for data preparation during extraction time will further improve responses, allowing analysts to make quick and effective decisions.

In the artificial intelligence realm, various techniques have been proposed and implemented to address the filtering problem. Ram [24] presents a theory of interestingness that serves as the basis for filtering and extracting data. However as argued before, for information that change rapidly and for the client with unpredictable needs, significant data processing is required for filtering tasks that involve changing data. Their approach cannot be customized to match user's changing requests.

We can also mention [25] explicit presents SIFTER system (Smart Information Filtering Technology for Electronic Resources), for filtering information and documents collected from the Internet and commercial databases. In [25] they highlighted the need for multiple

adaptation techniques to cope with uncertainties associated with changing interests of the user in information filtering environments.

Although ECA rules (triggers) have been explored and expressed as analysis rules for active data warehousing, putting such capabilities to use in a data preparation context still requires addressing. The need to envision new types of triggers, which would be needed for data preparation, was also addressed in [26]. Samtani [26] recognized that the set of constraints such as triggers could be used to solve existing data preparation problems. These constraints will act as sieves, which effectively filter data. For example in the case of data update, if some of the updates violate constraints then there is no need to update. Also changed data can be captured using constraints to check for validity.

Despite rapid advances in commercial tools and research proposals, most of the available solutions are relatively inflexible and limited in their features. We believe that a truly flexible and efficient solution to the data preparation problem requires further advances. The following section presents our motivating example to explain the need for a new approach to support data preparation processes in a dynamic and evolving data-warehousing environment.

III. MOTIVATION

Every meaningful active data warehouse application needs accurate data. A data warehouse is not of much use if results are invalid because the underlying data is inaccurate and inconsistent. Proper data preparation can cut preparation time and allowing analysts to produce quality analysis results. The current technologies for data preparation heavily focused on static snapshots of data. Ignoring changes may be accepted in certain scenarios, for example when it is not important for the analysis results to be current. Getting data quickly and efficiently to data warehouses for organizations such as hospitals, financial markets, banks, etc, is paramount. If currency, efficiency, and continuous access are required, then we believe that detecting and propagating changes will be the desired solution. Organizations are gunning for real time data preparation systems with operations such as load data as-you-go operations, filter data as-you-go operations, clean data as-you-go operations, etc. Many hospitals evolve their own data warehousing systems, often in an ad hoc fashion. This data tracking process is fairly well, but what hospitals are not particularly good at is tracing changes. Hospitals have been aware for sometime of the cases of injury or death caused by them, often due to missing or inaccurate information. Consider the following simple example

(Table1&2) where the business purpose of analysis is to identify all known illness and discarding missing data.

TABLE 1.
DATABASE TABLE CONTAINS THREE RECORDS WITH THREE ILLNESSES (CANCER, HEADACHE AND ME)

PNo	Surname	Forename	Date	Illness	Facts
2781	Frost	Jenny	20/10/05	Cancer	Jenny Frost diagnosed with cancer
2842	Jackson	Mic	21/10/05	Headache	Mic Jackson diagnosed with headache
2950	Smith	Fiona	25/10/05	ME	Fiona Smith diagnosed with ME

TABLE 2.
UPDATED TABLE CONTAINS FOUR RECORDS WITH THREE ILLNESSES (CANCER, HEADACHE AND ME)

PNo	Surname	Forename	Date	Illness	Facts
2781	Frost	Jenny	20/10/05	Cancer	Jenny Frost diagnosed with cancer
2842	Jackson	Mic	21/10/05	Headache	Mic Jackson diagnosed with headache
2950	Smith	Fiona	25/10/05	ME	Fiona Smith diagnosed with ME
3781	Frost	Jenny	20/12/05		Jenny Frost has no cancer

Note: Jenny Frost no longer has cancer

Incorrect data cleansing could result in damaging information in the data warehouse. In our example above if the action taken to clean data is based on previous business rule (purpose), the hospital could easily lose the fact that patient “Frost” no longer has cancer. What you believe about data and business purpose can change through time. This means that the data preparation should be described as a dynamic exercise rather than based on previous or one-off exercise. There has been an increased pressure for data preparation solutions for active data warehousing, which delivers the full patient history to the physician, not just information generated at some point in time (past or current) but also changes. This typically requires dynamic pre-processing methods. It is very important for physicians to have complete information about a patient. A leading hospital was recently forced to close down its test development program by the doctors, for it did not provide accurate/complete electronic medical records. The analysis of past patient information is a common approach in order to implement or test a decision. However patient information do not necessarily consist of data taken at some point in time,

last week's data or even last night's data is often not sufficient to react to certain situations. If data evolves fast then data preparation processes need to be capable of delivering up-to-date data for analysis. Preparing rapid changing data that land in a data warehouse is a major challenge. The dynamic nature can make it extremely difficult to gather, filter and cleanse it. The process of cleaning data in a dynamic environment has some identifiable problem areas:

- Managing uncertainties - require a high level of adaptability. For example, new information may be introduced as patient’s history changes. These changes are viewed as a source of uncertainty. Uncertainty in information systems is well-recognized problem [27].
- Managing iterative nature - the business purpose for which an analysis is carried out may be extended. This means a cleaning rule for one task may have to be revisited when another is under consideration. For this reason, there is a need to preserve cleaning rules for it to be available for subsequent cleaning rules.
- Lack of clear methodology

We are proposing a trigger cleaning approach where triggers are defined to manage the data cleaning process. Triggers are powerful and can provide an easy solution for many complicated problems from the automatic detection of errors and inconsistencies to the changing data in active data warehouse environments. Unfortunately, they are not used efficiently in the database environment. They take backseats to stored procedures, user-defined functions and static objects approaches. Even worse, the current data warehousing have neglected the role of triggers for data preparation. In dynamic environments, problems of data cleaning are not solved in a static world where one can identify what needs to be filtered and cleaned. Businesses require data preparation systems that can adapt to changes. As we have seen most of the previous works provide static or fixed and hardwired implementations not suitable for changing environments, which require more flexible and dynamic approaches. Businesses require systems that have predefined trigger concepts that can be easily modified to suite users needs. To support this flexibility our approach defines triggers with dynamic capabilities following the Dynamic Object Model (DOM) technology. The DOM technology, through specialization, is well placed to achieve our aim of flexibility. The DOM design has a structure that best suitable for handling changes [28]. It will allow trigger and its components (Event, Condition, Action) to be changed at runtime. We explain these components by using the framework specification presented in [2]. The UML diagram in Fig 1 describes trigger specification using the DOM technology. Trigger components

(Event-Condition-Action) are also described in the same manner as trigger in Fig 1.

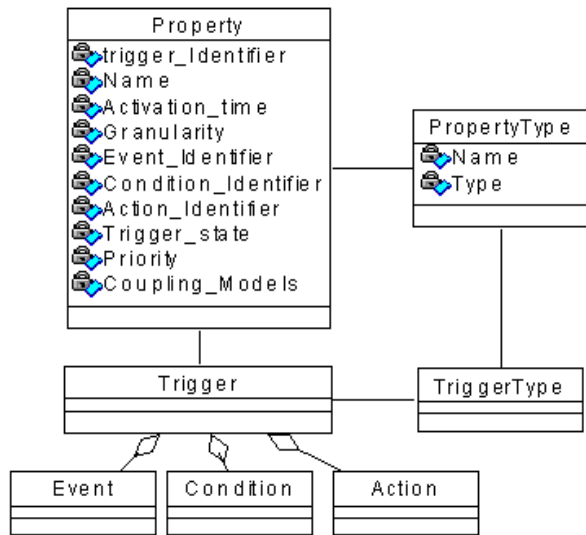


Fig.1. Dynamic Object Model of a Trigger (taken from [2])

IV. OUTLINE OF OUR PROPOSED APPROACH

A. Introducing Trigger Cleaning Approach (TCA)

TCA is based on the fact that action taken to clean data depends on business rules (purposes) that the data need to satisfy. These rules are expressed as triggers and separately stored in the database. Normally a trigger is defined in the Event-Condition-Action (ECA rule) fashion. Given this structure of a trigger language, a simple trigger can be generated automatically when supplied with necessary attributes. An event part of a trigger can be used to perform operations such as updating, inserting, loading, etc. Complex predicates, functions or checks can be contained in the condition part for checking errors, duplicate data, inconsistency, missing values, etc. It is necessary to specify the condition criteria for accepting that values are cleaned for the specified purpose. The condition criteria can apply to individual values as well as set of values. The condition criterion helps to identify the problem or problems. Once identification has been made then some transformation rules have to be implemented. The transformation rule specifies the action to be taken to transform data. This could be omitting a record, substituting a value, confirmation operations such as sending message alerts or other database operations, etc. The action part of a trigger will be activated when the condition is satisfied. Well-developed triggers are effective in data cleansing processes for identifying errors, duplicates, missing values, transformation of data, etc. A cleaning trigger for one analysis or business purpose may be destructive to a subsequent analysis as

demonstrated in (section III). Triggers are separately stored in the database away from preprocessed data. This is not only easy to change triggers but also can be reused without modification if they possess similar business rules.

B. Trigger Cleaning Process

The trigger-cleaning diagram (Fig. 2) shows a high-level overall data-cleaning model. It starts with preprocessed (original) data with a variety of errors. Preprocessed data is checked against predefined triggers in the database with the objective of obtaining correct and consistent data, and then are loaded into data warehouse. The effectiveness of the cleaning process depends on the triggers (rules) defined in the database. The definition of a trigger for cleaning process involves the following:

1. Establishing preprocessed data for cleaning (Event)
2. Specification of acceptance criteria (Condition)
3. Transformation rules to give a cleaner dataset (Action)

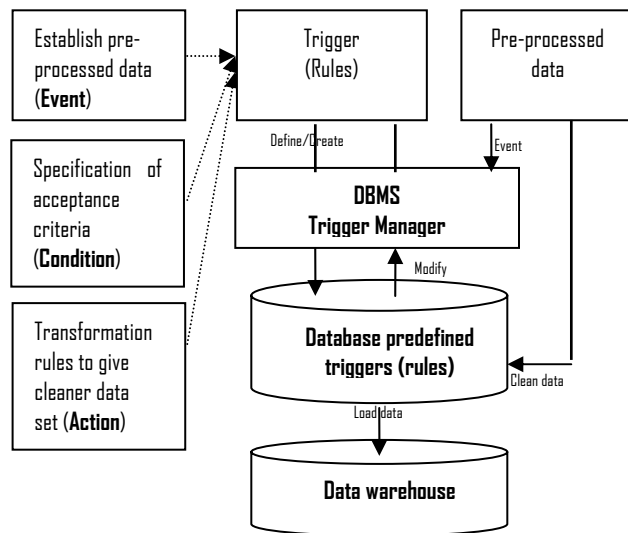


Fig.2. Trigger cleaning model

TCA is not only concerned with cleaning data but also recording of triggers (rules) taken in cleaning the data. This approach consists of two stages namely the trigger processing stage and the data processing stage. The trigger processing stage is for creation and manipulation of triggers and their components. The effort is channeled into development of trigger management system, which is responsible for trigger creation and manipulation functionalities. The data processing stage is the actual data-cleansing phase. Fig.

3 provides a systematic approach for a data cleansing process in an active data warehouse environment.

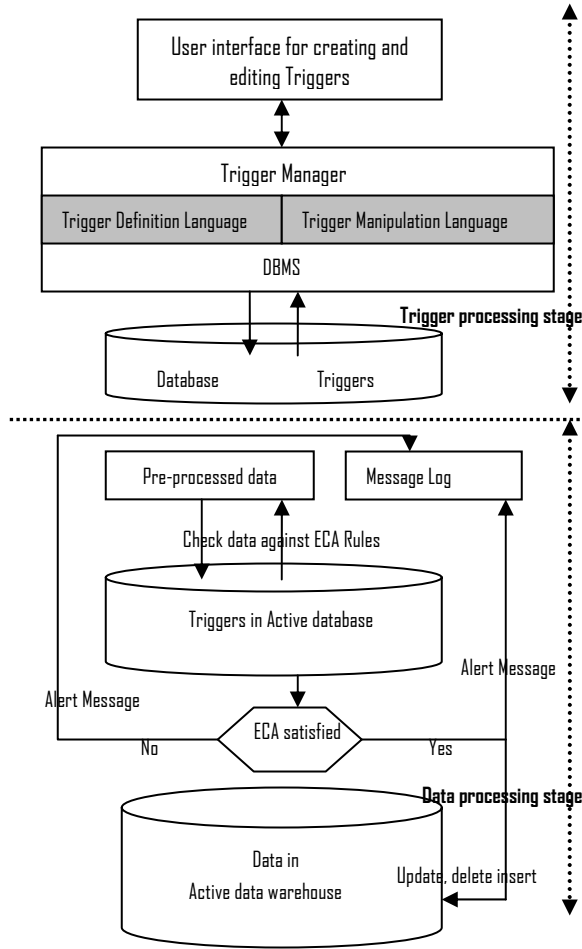


Fig.3. Trigger based cleaning framework

1) *Trigger Processing Stage (Semantic data preparation)*

Domain knowledge has been identified as one of the main ingredients for successful data cleaning [29]. This stage supports the issue of knowledge creation and representation for data cleaning process. Triggers and their ECA components are created and stored in the database using the Trigger Definition Language part of the Trigger Manager that will be added on top of a database management system (DBMS). Triggers are defined in advance in order to do current and later or future checks through the lifespan of a database. The Trigger Manipulation Language also forms a part of the Trigger Manager. Trigger Manipulation Language is responsible for trigger manipulation operations such as selection, insertion, update, etc. Furthermore, since our triggers have dynamic capabilities, a trigger and the three components (Event-Condition-Action) are easily modified to suite users needs.

2) *Data Processing Stage (Syntactic data preparation)*

Since the trigger manager is part of the DBMS, fetching and cleansing processes is done automatically. Clean data is available to the data warehouse database straight away. Triggers are fired when preprocessed data is fed into the database. An Event part will be activated and conditions will be checked before changes are made into the database. When an event part is true, the preprocessed data is first evaluated by the condition part of a trigger to detect any errors, inconsistency, missing values, etc. The action part of the trigger will be fired when the conditions are satisfied.

C. *Simple Trigger Cleaning Example*

To illustrate our TCA, we consider the following dataset on the respiratory illness. Briefly this data comes from clinical study in which the age of patients with respiratory illness was examined. The analytical need (business purpose/rule) was to provide a number of patients over 50yrs with respiratory illness.

TABLE3. PATIENTS WITH REPOSITORY ILLNESS

Patient Name	Sex	Age
A	F	51
B	F	60
C	M	2

← Dirty Data

The following trigger will be stored into the database to clean dirty data (Table3).

Trigger name – Over50respiratorypatient

Event – Insert

Condition – age < 50

Action – Delete row

Through time the analytical need was changed to provide an estimate of respiratory patients over 50yrs as well as those less than 5yrs. Since triggers and their ECA components are treated as dynamic objects. Similar to data in Relational databases, triggers are not only created and stored in a database but also manipulated. The cleaning process will be very easy. We just need to change the above trigger into the following:

Trigger name – Over50respiratorypatient

Event – Insert

Condition – 5 > age < 50 (Changed)

Action – Delete row

Using our TCA will allow users to create triggers:

- For detecting and removing inconsistency data entry errors usually typos, phonetic error, misspellings, naming conflicts, structural conflicts, duplicates (redundancy) and contradicting records, etc
- For transforming data as needed to conform and standardize data definitions (attribute values should be converted to a consistent and uniform format, e.g. using identical criteria for "customer")
- For finding and filling missing information and dummy data, etc

Moreover user can process queries such as:

- Find and modify trigger (rule) that needs changing
- Find all triggers (rules) that perform certain business purpose/ rule
- Retrieve and update a certain trigger e.g. trigger that transforms for example American date format to British data format, etc

V. FUTURE WORK AND CONCLUSION

In this paper we have presented a trigger mechanism for data cleaning based on DOM approach. The advantages of specifying triggers in the DOM were identified in [2]. The benefit of using this mechanism is the integration of active or dynamic capabilities to data cleaning process. Our main contribution is to enhance data preparation operations through triggers. The TCA provides an efficient knowledge management for data cleaning tasks through triggers. Our semantic process is a timely and continuous data preparation (day to day) while the database is available. Considering this approach, it is possible to achieve the following:

- Fast modification of triggers and their ECA components – using Dynamic objects we can modify components at run time hence run time modification of data cleaning processes.
- Reusability of trigger and their ECA components – the components can be shared among applications. Also since they are stored in the database we can avoid redundancy.
- Better organization of trigger and their ECA components – components can be enlarged to cope with later extension.

The implementation issues of the proposed model are research areas that have a lot of work to be done. As part of future work, we will focus on implementation issues. The concepts discussed in this paper can be implemented in any active Object-Oriented database system. We are currently implementing the concepts presented in this paper on an object-oriented database

Objectivity/DB. Schema objects such as triggers and their components are added into Objectivity/DB database using the c++ implementation language. The user interface is also considered. The implementation of the user interface depends on the structure used for storing triggers and their components (Trigger Manager Schema). The task of the user interface is to support navigation through a Trigger Manager. The user interface will display all the constructs supported by the Trigger Manager to allow the inspection of triggers and their components. It may also return dependency information about triggers and data stored in a database. The user interface will also offer editing facilities for creating, updating and deleting triggers and their components from the database. In addition for any application that changes, data is modified. A data warehouse must efficiently handle expired (outdated) data. Techniques are needed for specifying a currency of data requirements in a warehousing environment and for ensuring that outdated data is automatically and efficiently removed from the warehouse.

To conclude, the work presented in this paper is a possible step forward towards enhancing data preparation with semantic aspects and shorten the time for data cleansing processes in active data warehouse environments.

ACKNOWLEDGMENT

It is a pleasure to acknowledge the support and help provided by Alex Tarry of London Metropolitan University.

REFERENCES

- [1] Xiaou Li., Mann J M, and Chapa S V, 2002, A Structural Model of ECA Rules in Active Database. Proceedings of the Second Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, Lecture Notes In Computer Science; Vol. 2313, Pages: 486 – 493, ISBN: 3-540-43475-5
- [2] Ezekiel, K. and Marir F, 2003. A conceptual model for managing knowledge represented as triggers in active databases”: 4th European Conference on Knowledge Management, pp. 323-333
- [3] Abiteboul S, Tova S C, and Zohar S, 1999, Tools for Data Translation and Integration. IEEE Data Eng. Bulletin, Volume 22, pp3-8
- [4] Chaudhuri S. and Dayal, U, 1997, An Overview of Data Warehousing and OLAP Technology, SIGMOD Record, 26(1): 65-74
- [5] Galhardas. H, Florescu D, Shasha D, Simon E, 2000, declaratively cleaning your data using AJAX, Technical University of Lisbon, JNICT PRAXIS XX1, Portugal
- [6] Rahm. E, and Do. HH, 2000, Data Cleaning: Problems and Current Approaches, IEEE Bulletin of the Technical Committee on Data Engineering, Volume 23 No. 4
- [7] Galhardas, H. Florescu D, Shasha D, Simon E, and Saita C A, 2001, Declarative Data Cleaning: Language, Model and Algorithms, Proceedings of the 27th International Conference on Very Large Data Bases, Italy, Pages: 371 - 380

- [8] Bohn, K, 1997, "Converting data for warehouses", DBMS, Volume 10, Issue 7, Pages: 61 – 66, ISSN: 1041-5173, Miller Freeman, Inc. San Francisco, CA, USA
- [9] Maletic, J, and Marcus, A, 2000, Data Cleansing: Beyond Integrity Analysis, Proceedings of The Conference on Information Quality (IQ2000), Massachusetts Institute of Technology, Boston, MA, USA, pp. 200-209
- [10] Vijayshankar, R and Hellerstein, J M, 2000, An interactive framework for data cleaning, EECS Department, University of California, Berkeley, UCB/CSD-00-1110
- [11] Raisinghani, V T, 1999, Cleaning methods in data warehousing, Seminar Report
- [12] Galhardas H, Florescu D, Shasha D and Simon E, 2001, "Improving Data Cleaning Quality Using a Data Lineage Facility" *Book title: Design and Management of Data Warehouses*, pp. 3
- [13] Hernandez, M A and Stolfo, S, 1998, Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem, *Data Mining and Knowledge Discovery* Volume 2, pp.9-37
- [14] Rahm, E and Do, HH, 2000, Data Cleaning: Problems and Current Approaches, IEEE Bulletin of the Technical Committee on Data Engineering, Volume 23 No. 4
- [15] Monge A E and Elkan, P C, 1997, An Efficient Domain-independent Algorithm for Detecting Approximately Duplicate Database Records, Proceedings of ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, pages 23--29
- [16] Clear J, Dunn D and Harvey B, 1999, NonStop SQL/MX primitives for knowledge discovery, ACM SIGKDD international conference on Knowledge discovery and data mining, Pages: 425 – 429, ISBN: 1-58113-143-7
- [17] Cooley R, Mobasher B and Srivastava J, 1997, Web Mining: Information and Pattern Discovery on the World Wide Web, Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), Newport Beach, CA: 558 – 67
- [18] Cooley, R, Mobasher B, and Srivastava J, 1999, Data Preparation for Mining World Wide Web Browsing Patterns, *Knowledge and Information Systems vol1*, pp. 5-32
- [19] Zhang S, Yang Q, and Zhang C, 2003, "Data preparation for data mining", *Applied Artificial Intelligence* Volume 17:375-381
- [20] Cooley, R, 2003, "The use of web structure and content to identify subjectively interesting web usage patterns", *ACM Transactions on Internet Technology (TOIT)*, pp.93-116
- [21] Buchner, A G, 1998 "Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining", *SIGMOD Record* Volume 27 pp.54-61
- [22] Machado, L D S and Becker, K, 2003, Distance Education: a Web Usage Mining Case Study for the Evaluation of Learning Sites, Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)
- [23] Pyle, D, 1999, *Data preparation for data mining*, Morgan Kaufmann Publishers, USA, ISBN: 1-55860-529-0
- [24] Ram, A, 1991, Interest-based information filtering and extraction in natural language understanding systems, Bellcore Workshop on High-Performance Information Filtering
- [25] Mostafa J, Mukhopadhyay S, Lam W and Palakal M, 1997, A multilevel approach to intelligent information filtering: model, system, and evaluation, *ACM Transactions on Information Systems*, pp.368-399
- [26] Samtani, S, Mohania M, Kumar V and Kambayashi Y, 1998, Recent Advances and Research Problems in Data Warehousing, Proceedings of the Workshops on Data Warehousing and Data Mining, Pages: 81 – 92, ISBN: 3-540-65690-1
- [27] Motro, A and Smets, P, 1996, Uncertainty Management in Information Systems: From Needs to Solution. Kluwer Academic Publishers, Boston, ISBN 0-7923-9803-3
- [28] Riehle D, Et al, 2000, Dynamic Object Model, In Proceedings of the 2000 Conference on Pattern Languages of Programming - PLoP
- [29] Maydanchik, A, 1999, Challenges of efficient data cleansing, <http://www.dmreview.com/editorial/dmreview>