An Integrated Conceptual Model for Temporal Data Warehouse Security

Marwa S.Farhan
Faculty of Computers and Information, Helwan University, Egypt
E-mail: Marwa_salah@hotmail.com

Mohamed E. Marie
Faculty of Computers and Information, Helwan University, Egypt
E-mail: mohamedmarie@yahoo.com

Laila M. El-Fangary & Yehia K. Helmy
Faculty of Computers and Information, Helwan University, Egypt
E-mail: {lailaelfangery@gmail.com, yhelemy@yahoo.com}

Received: May 23, 2011 Accepted: June 10, 2011 doi:10.5539/cis.v4n4p46

Abstract

In the past few years, several conceptual approaches have been proposed for the specification of the main multidimensional (MD) properties of the data warehouse (DW) repository. However, most of them deal with isolated aspects of the DW and do not provide designers with an integrated and standard method for designing the whole DW life cycle (ETL processes, data sources, DW repository and so on).Some approaches are depending on specific platform or neglecting important issues in DW design life cycle. Extraction-transformation-loading (ETL) processes play an important role in data warehouse architecture because they are responsible of integrating data from heterogeneous data sources into the DW repository. It is vitally important to consider security requirements from the earliest stages of the development process. This paper proposes a conceptual model to refresh data warehouse by (insert, update, delete) data using ETL processes and considering DW security requirements. Firstly, the proposed ETL model is based on Unified Modeling Language (UML), which allows us to accomplish the conceptual modeling of ETL processes .secondly; this part focuses on how to integrate the proposed ETL model with the DW model. The proposed DW model depends on the Model Driven Architecture (MDA). MDA is a standard framework for software development that addresses the complete life cycle of designing, deploying, integrating, and managing applications by using models in software development. This paper proposes an integrated conceptual model for addressing temporal data warehouse security requirements (CMTDWS). One key advantage of our model is that we accomplish the conceptual modeling of secure temporal DWs independently of the target platform where the DW has to be implemented, allowing the implementation of the corresponding TDWs on any secure commercial database management system.

Keywords: Data warehouse, Temporal data warehouse, ETL, DW security, Conceptual modeling, UML

1. Introduction

ETL processes are responsible for the *extraction* of data from heterogeneous operational data sources, their *transformation* (conversion, cleaning, normalization, etc.) and their *loading* into DWs. ETL processes are a key component of DWs because incorrect or misleading data will produce wrong business decisions, and therefore, a correct design of these processes at early stages of a DW project is necessary to improve data quality. Several proposals have been defined for the conceptual modeling of ETL processes. Data warehouse refreshment is a complex process comprising many tasks, such as extraction, transformation, integration, cleaning, key management, history management, and loading .To fulfill decision support needs; data warehouses can use data from many internal and external sources. Reconciling the differences among data sources is a significant challenge. As data sources change at different rates, the determination of the time and content to refresh can be a significant challenge, refreshing a data warehouse is the most significant activity in the operation of a data warehouse. As far as we are aware, this paper proposes the first integrated conceptual model for addressing temporal data warehouse security requirements (CMTDWS).This model is the first model which combines ETL modeling specifications with temporal data warehouse (TDW) modeling specifications in one integrated model. Our model based on MDA architecture and uses UML notations in modeling a conceptual model to refresh data warehouse (insert, update, delete) using ETL processes and considering DW security requirements. One of the best advantages of our proposal is that we model secure TDWs from a conceptual perspective which is totally independent of implementation details. The main relevant advantage of this model is that it uses the UML, a widely accepted object-oriented modeling language, which saves developers from learning a new model and its corresponding notations for specific MD modeling. This paper is organized as follows: In Section 2, we present the related work. Section 3 offers a very clear and high-level overview of ETL processes. Section 4 explains in detail the ELT concepts and processes that used to refresh DW considering temporal DW aspects. Section 5 extends the MDA model which is presented in (E.Fern´andez-Medina, J.Trujillo and M.Piattin, 2009) to model the temporal DW (TDW) and integrate the DW proposed model with the ETL proposed model to representing the TDW aspects and DW security requirements. Finally, we encompass the main conclusions and future work in section 6.

2. Related Work

This section divides the related work according to the three main research topics covered by this paper: *ETL modeling*, *Security and access control models for DW*, and *Temporal data warehouse*. The paper focuses on the conceptual modeling specifications in these topics.

2.1 ETL Modeling

A conceptual modeling of ETL processes in data warehouses using UML is proposed in (P. Vassiliadis, A. Simitsis, P. Georgantas,M. Terrovitis, and S. Skiadopoulos,2005). The necessary mechanisms for an easy and quick specification of the common operations defined in the ETL processes also are provided. The conceptual modeling of ETL Process is discussed in (P. Vassiliadis, A. Simitsis, S. Skiadopoulos, 2002)(Y. Feng, H. Li, D. Agrawal and A. Abbadi,2005). On all these researches the authors discussed ETL with UML or security with UML. But they have not explored an integrated model for addressing security requirements in ETL processes with DW requirements.

2.2 Security and access control models for DWs

In the relevant literature, we find several initiatives researches for the inclusion of security in DWs. Many of them are focused on interesting aspects related to access control, multilevel security, the applications of these aspects to federated databases, applications using commercial tools and so on. For instance, authors present a prototype model for DW security based on metadata, which defines different user groups, and a different view of data for each user group. These researches are interesting, but in a real situation, grouping users and defining a different view for each group is not enough, being necessary to combine groups and to specify complex confidentiality security constraints. Additionally, there are some interesting proposals that try to define an authorization model for DWs (Weippl, O. Mangisengi, W .Essmayr, F. Lichtenberger and W. Winiwarter,2001),(T. Priebe and G. Pernul,2000),(T. Priebe and G. Pernul,2000), but they mainly deal with OLAP (Online Analytical Processing) operations accomplished with OLAP tools. Therefore, they are not conceived to be integrated in the MD modeling as part of the DW design process. Furthermore, from our point of view, we should consider basic security aspects of DWs by means of a conceptual model from the early stages of a DW project, and then more specific security rules can be defined for particular groups of users by OLAP tools or any other analysis tool, but being consistent with the main general security rules defined for the DW.

2.3 Temporal Data warehouse

From the point of view of a data warehouse system, collecting and receiving information from source systems is crucial for all subsequent business intelligence applications. Incoming information can generally be classified into two types: (1) the state-oriented data and (2) event-oriented data or transactional data, which contains information about the change performed by processes on the instances of information objects. Data updates are important issues for temporal data warehousing (N. Rahman, 2008).The data warehouse refreshes have been a research topic for more than a decade. The research is mostly related to storing and maintaining the current state of data. Discarding updates between two refresh points of time when performing periodic complete reloads leads to a loss of transaction lineage (A. Vavouras, S. Gatziu and K .Dittrich, 1999). On the way towards achieving the goal of a full-fledged active data warehouse it becomes more and more important to provide data with minimal latency.

Kimball (R. Kimball, M. Ross and R. Merz, 2002) has introduced three slowly changing dimension (SCD) types to track changes in dimension attributes. In the SCD Type 1, the changed attribute is simply overwritten to reflect the most current value thus does not keep the historical changes. The SCD Type 2 creates another (dimension) record to keep trace the changed attributes, but could not keep the old value both before and after the change. For this purpose, the SCD Type 3 uses the current value and previous value which is not an appropriate method for unpredictable changes. There has been little research work done to date on the temporal view maintenance problem (J. Yang and J. Widom, 1998) and most of the previous research ignores the temporal aspects of data warehousing (R. Bruckner and A. Tjoa, 2002).Temporal data warehouses raise many issues including consistent aggregation in presence of time-varying data, temporal queries of multidimensional data, storage method, and temporal view materialization (E. Malinowski and E. Zimányi, 2006). The temporal aggregation problem was studied in (J. Yang and J. Widom, 2001) (Y. Yufei Tao, D. Papadias and C. Faloutsos, 2004) (D. Zhang, A. Markowetz, V. Tsotras, D. Gunopulos and B. Seeger, 2001) to address the challenges of temporal data. Much research is now being done to improve the efficiency of range aggregate queries in a temporal data warehouse (A. Tjoa, M. Nguyen, et al,2007) (Y. Feng, H. Li, D. Agrawal and A. Abbadi,2005) (N. Rahman,2008).

3. Extraction-Transformation-Loading (ETL)

The section presents the first part of the proposed model (CMTDWS).This part uses ETL processes to refresh data in TDW by insert, update and delete data .First, we explain ETL processes in details. Second, we introduce DW refreshment aspects**.**

3.1 ETL Processes

Data from the operational systems are usually specified in different schemas and have to be extracted and transformed to collect them into a common DW repository***;*** in an ETL process, the data extracted from a source system pass through a sequence of transformations before they are loaded into a DW. The design of an ETL process is usually composed of seven tasks

1-Filter: The Filter mechanism filters unwanted data and verifies the correctness of data based on constraints.

2- Join: The Join mechanism is used to join two data sources related to each other with some attributes *(defined by means of a restrictive condition)*

3-Loader: The loader mechanism loads data into the target of an ETL process such as a dimension or a fact in a DW. Every ETL process should have at least one loader mechanism. ***Two operation modes are supported in the Loader*** *(i) Free loading:* the Loader mechanism does not verify any constraint as the target applies its own constraints to the new data. (ii)*Constrained loading:* the designer can apply primary and foreign key constraints during the loading process. Moreover, the designer can also define how to manage existing data in the target; we use the constraining loading in CMTDWS.

4-Incorrect: The Incorrect mechanism is used to reroute bad or discarded records and exceptions to a separate target. In this way, the DW designer can track different errors. This mechanism can only be used with the Filter, Loader, and Wrapper, because these mechanisms constrain the data they process.

5- Merge: The Merge mechanism integrates two or more data sources with compatible attributes. Two data sources are compatible as long as both of them contain a subset of the attributes defined in the target; the attributes used in the integration must have the same names in all the data sources.

6- Wrapper: The Wrapper mechanism allows us to define the required transformation from a native data source into a record based data source.

7- Surrogate: The Surrogate mechanism generates unique surrogate keys. Surrogate key assignment is a common process in DWs, employed in order to replace the original keys of the data sources with a uniform key.

We will focus on some of ETL processes to refresh the DW by executing the data manipulation language (DML) operations which are insert, delete and update. The update of DW means delete the old value then insert the new one; and also we model the insert operations in ETL model. The temporal data warehouse aspects are used to execute these operations; as it is explained in the next section.

3.2 DW Refreshment Aspects

Updating data warehouses with temporal data is a mechanism for storing the lineage of data in the tables; it captures all changes made to data row overtime. CMTDWS considers some constraints to mange (insert/update/delete) data in SCD, these constraints are:

–Non-temporal dimension: Natural key must not exist in the target object on insert; no update or delete operations are allowed in this dimension.

–Temporal dimension (SCD): Natural key must exist in the target object on update and delete. Deleted rows are not physically deleted; they are labeled to exhibit expiration instead. Updated rows are handled by expire the existing row and insert the new version of the row. *The constraints to handle SCD are:*

Time: We use valid time (start time and end time) to represent the time period of the row in SCD. In update process the start time (SVT) of the new row(current version) is end time (EVT) of the previous row (old version). In the case of delete operation, we will not delete the row we just expire that row by put the time of deletion as the end time of the deleted attribute.

Versions: use surrogate key to differentiate between versions of the same key in the temporal data which exists in SCD.

4. ETL Proposed Model

In an ETL process, the data extracted from a source system pass through a sequence of transformations before they are loaded into a DW. We present our ETL modeling proposal that allows the designer to check the DML processes in ETL to refresh the DW and we will consider the security requirements in the DW model itself later in the next sections. CMTDW focuses on two ETL processes which clarify our idea.

4.1 Filter Process

Filter process checks the following constraints:

Data source constraints: We propose to make a list to store data sources, Figure1, shows the data source class and its attributes, we use UML class notation to represent data source. Every data sources has an allowed privileges (Insert, delete, update) so we check the data sources privileges to see if it is authorized to perform the DML operations on data in DW or not ,we reject any un authorized data source before extract data from it this procedure save a lot of time and cost .

Datasource_Id: data source code.

Datasource_Name: data source name.

Action: it represents the allowed privileges that data source is allowed to do on DW dimensions. For example, we use this attribute to check if there is any action (insert, update, delete) on any data in data source which is not one of its allowed privileges. In this way we prevent any unauthorized data source from changing data and support the DW security aspects.

Dimensions constraints:

• **PK**(source attributes): defines the attributes of the source data that define a unique record in the target. This information is used for both constraining the loading process and detecting the old data that should be updated.

• **FK**(target attributes; source attributes): defines the attributes of the source data that should previously exist in a target.

The CMTDWS considers two situations:

1**. Change affect non temporal dimension:** just insert new rows.

2. **Change occurs if SCD is affected, change could be (I/U/D):** in this case we propose creating a new table in ETL which called (TransEvent) table .This table contains the transactions that happened science the last refreshment of DW- *it Just store the changed data (events)* - each record in the TransEvent table corresponds to a change value of one dimension attribute in which we are interested in.

In figure2, we use UML class diagram to represent TransEvent, It shows the TransEvent class attributes: *Key:* is the primary key of the changed attribute, *AffectedAttribute:* the changed attribute from the last refresh *Action:* the action that occurred in the affected attribute, it can be (insert/update/delete).*Time:* the transaction time (TT) when the action occurred, It is useful to know the sequence of the changes on attribute. *Datasource***:** the source in which the change occurs**.**

Figure 3 explains the temporal data warehouse (TDW) refreshment cycle, In each refresh cycle the filter process extracts the changed data in SCD which occurred since the last refresh (i.e., those records which do not appear in the TransEvent). The filter checks the previous constraints depends on the action attribute (I/U/D) and key attribute .If there are any invalid attributes we can add it to another table (Report Table) through the Incorrect mechanism, we propose to create a list of reasons and we put the reason code in reason attribute in Report table to explain why the data is rejected.

Figure 4 shows that the valid attributes have a new surrogate key in the case of (I/U) for each valid record this step is done through the surrogate mechanism (SK).For example, in product dimension (temporal dimension) assume that the changed attribute is (Color) .If action attribute is update, we need to add a new surrogate key for the new record, so we extend the primary key of the target table with a new Sk.

We consider several cases which can occur when we refresh data warehouse, these cases and the solution to each case is introduced in table1.

4.2 Load

A temporal data warehouse provides a consistent view of data for customer queries while data is being loaded into the same table being queried. It provides mechanism to harmonize (aggregate and synchronize) data based on time slicing. The business needs for several different time-varying data can be met using a temporal data warehouse. We consider the temporal data warehouse requirements in our model so as we mentioned before the DW data is neither modified nor deleted. Thus, TT (transaction time) in TDWs represents the time when data was loaded into a TDW. This time is called in our model data warehouse loading time (DWLT). DWLT can differ from TT or VT (valid time) of source systems due to the delay between the time when the changes occurred in source systems and the time when these changes are integrated into a TDW. DWLT is important especially in active DWs and in creating TDWs from non-temporal sources (J. Yang and J. Widom, 1998). In the load process we add the DWLT to the SCD to represent the time when the data is loaded in DW; this DWLT represents the (Start_time and end_time) of each row in SCD.

In figure 5, CMTDWS uses the UML activity diagram to representing the ETL model .A summarization of the ETL processes which used in CMTDWS is presented in the following steps:

1-Source Authentication: It identifies the data sources and prepares list of data sources.Add the new data source, if any, and check the data source allowed privileges.

2-Extractraction: we extract the modified data only, we do this by comparing the old TransEvent table (from the previous cycle) with the new coming data the extracting the new incoming data from new DW refresh cycle and store it in the TransEvent table.

3-Filtering process: Enforce the defined constraints (e.g. primary key, foreign key) and the user constraints as (insert. update, delete) constraints.

A. SCD constraints:

Check the operations that assigned in (***O attribute***) in the DW's SCD dimension to check the data validity.

B. Non temporal dimension constraints:

Check (primary data, foreign data) and any user defined constraint.

4. Incorrect Process: the invalid data is stored in report table**,** to send the reason of the invalidations to the data source.

5. Surrogate process: generates a new surrogate key to reflect the version number in the SCD.

6. Loading Process: Loads the data to the data storage area (DSA), in the case of (I/U/D) the SCD has a time attribute (start_time, End_time), we consider data warehouse loading time (DWLT), it differs from TT or VT of source systems due to the delay between the time when the changes occurred in source systems and the time when these changes are integrated into a TDW (in SCD).

5. Extend the MDA Model

The section presents the second part of CMTDW, it is based on the MDA structure and the model which proposed in (E.Fern´andez-Medina, J.Trujillo and M.Piattin, 2009) (R. Villarroel, E. Fernández-Medina, M. Piattini, J. Trujillo, 2007).We will extend this model to cope with our approach by extending the model privileges (read) to make it possible to specify security aspects in the crucial ETL processes for TDWs and considering other privileges such as delete, insert and update to represent TDW aspects.

5.1 MDA structure

MDA is an Object Management Group (OMG) standard that addresses the complete lifecycle of designing, deploying, integrating, and managing applications. MDA separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. Thus, MDA encourages specifying a Platform Independent Model (PIM) by using any specification language – (UML). Then, this PIM can be transformed into multiple Platform Specific Models (PSM) in order to be executed on a concrete platform by transforming this PSM into the corresponding Code. In Figure 6, we can see the general overview of the MDA architecture, in which one PIM can be transformed into different PSMs, and one PSM can be transformed into different Codes according to the concrete platform where the system is to be implemented. Alternatively, horizontal transformations (normally called bridges) can also be defined in order to transform models at the same level.

5.2 Secure Temporal Data Warehouse Model

This paper focuses on the conceptual model which is called PIM in the MDA architecture, we extend the PIM model which proposed in (E.Fern´andez-Medina, J.Trujillo and M.Piattin,2009)(R. Villarroel, E. Fernández-Medina, M. Piattini, J. Trujillo,2007) .PIM corresponds with an extension of the UML in which the information is clearly organized into secure facts and secure dimensions is proposed in (R. Villarroel, E. Fernández-Medina, M. Piattini and J. Trujillo,2006).It allows us to classify both information and users in order to represent the main security aspects in the conceptual modeling of DWs. Security information is defined for each element of the model (SFact, SDimension, SFactAttribute, etc.) by specifying a sequence of security levels, a set of user compartments and a set of roles. Moreover, the constraints (AuditRule, AuthorizationRule and SecurityRule) are modeled by using UML notes. These constraints are defined by following the syntax of the ARs, AURs and SIARs' rules from the ACA model ,more details in (R. Villarroel, E. Fernández-Medina, M. Piattini and J. Trujillo,2006) (E. Fernández-Medina, J. Trujillo, R. Villarroel, M. Piattini,2006) (E. Fernández-Medina, J. Trujillo, R. Villarroel, M. Piattini,2007). The CMTDWS extends this model to represent TDW aspects integrated with DW security requirements, so we add to each SCD in DW set of the allowed operations as we will explain later.

CMTDWS uses the UML class diagram to represent fact and dimensions in the DW part, we use UML stereotypes to represent SCD (<<TD>>) which represents temporal dimension and uses OCL (Object Constraint language) constraints to represent security level and allowed operations as it is showed in figure 7. The OCL constraints on SCD dimension are:

SL: Security level indicates the clearance level of the user such as Top Secrete(TS),Secret(S),Confidential(C),and unclassified(U),where TS>S>C>U.

O: represents the allowed operations on the SCD attributes which are (Insert (I), Delete (D), Update (U)) .This constraint is checked in ETL also to accept or reject data according to the allowed operations on the dimension, thereby the model supports the data warehouse security aspects in ETL by which we guarantee the data security and validity before loading data in DW. It is the most important advantage of CMTDWS because it saves time, cost and supports data integrity.

The attributes of SCD (product dimension) are:

Sk: number of product version (1,2,.)

P_id: primary key (natural key)

Other attributes: (p_name,….)

Start time: is the time when the data is loaded (DWLT)

End time: in case of (D,U) ,the start time of the new value of the new version of the attribute, the value is of end time in the new row is *till now*.

In figure 8 extends the user profile which is proposed in (E.Fern´andez-Medina, J.Trujillo and M.Piattin,2007) ,CMTDWS extends the user profile by adding time to user status ,the time period changed every time the user has new privileges or new roles. We check the security rules and also check time period every time the user wants to access data in DW. We can keep historical data as in SCD and use any type of SCD handling cases (overwrite, new record, etc). The main procedure of the CMTDWS to secure data extraction process and TDW in DW is as follows:

1-User Profile: we add time to check privileges (security aspects) considering if it’s allowed in this period of time or not.

2. Extending SCD to represent TDW

A. *Add (O attribute)*: in SCD to set the allowed privileges (I, U, D) to check these constraints against the extracted data in ETL to verify data validity.

B. Add time (Start time, End time) and handle it as we explained before.

C. Add surrogate key which is generated in ETL stage to differentiate between the versions in SCD when we refresh DW with new data which is added in (I/U/D) operations.

6. Conclusion and Future work

In this paper we have presented CMTDW which is an integrated model that proposes how to integrate ETL processes and temporal data warehouse security requirements together in one model. Our model divided to two parts which are: 1 ETL part, 2: Data warehouse part. In ETL part, we handle the insert, update and delete operations considering temporal data warehouse aspects and security issues. In DW part, we handle temporal data warehouse aspects and security issues. We also explain the needed procedures to integrate the ETL and DW in one integrated model. The most important advantage of CMTDWS is that it supports the data warehouse security aspects in ETL by this we guarantee the data security and validity before loading data in DW, the proposed model saves time, cost and supports data integrity. One key advantage of our model is that we accomplish the conceptual modeling of secure temporal DWs independently of the target platform where the DW has to be implemented, allowing the implementation of the corresponding TDWs on any secure commercial database management system. Another relevant advantage of this model is that it uses the UML, a widely accepted object-oriented modeling language, which saves developers from learning a new model and its corresponding notations for specific MD modeling.

Our immediate future work consists of providing both the definition and the implementation of the Query/Views/Transformations (QVT) relations in order to establish a transformation between the conceptual model (PIM) and the logical model (PSM) levels of the CMTDWS. We shall implement the extended CMTDWS to some commercial tools in order to elicit security requirement for DWs at the business level and we shall accomplish this with the MDA in order to cover the whole lifecycle of DWs, Our next steps include the implementation and improvement of the proposed model and subsequent case studies.

References

A. Vavouras,S. Gatziu and K .Dittrich. (1999). Modeling and Executing the Data Warehouse Refreshment Process, International Symposium on Database Applications in Non-Traditional Environments (DANTE '99),pp.66-73.

A. Tjoa, M. Nguyen,et al. (2007). An approach towards an event-fed solution for slowly changing dimensions in data warehouses with a detailed case study, *Data&Knowledge Engineering,* vol 63, Issue 1, October 2007,pp. 26-43.

D. Zhang, A. Markowetz, V. Tsotras, D. Gunopulos and B. Seeger. (2001). Efficient computation of temporal aggregates with range predicates, In Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp.237 - 245.

E.Fern´andez-Medina, J.Trujillo and M.Piattin. (2007). Model Driven multidimensional modeling of secure data warehouses. *European Journal of Information Systems.* 16, pp.374–389.

E.Fern´andez-Medina, J.Trujillo and M.Piattin. (2009). A UML 2.0 profile to define security requirements for Data Warehouses, *Computer Standards & Interfaces - CSI ,* vol. 31, no. 5, pp. 969-983.

E. Malinowski and E. Zimányi. (2006). A conceptual solution for representing time in data warehouse dimensions, In Proceedings of the 3rd Asia-Pacific conference on Conceptual Modeling (APCCM 2006), pp.45-54.

E. Weippl, O. Mangisengi, W .Essmayr, F. Lichtenberger and W. Winiwarter. (2001). An authorization model for data warehouses and OLAP. In Proceedings of the Workshop on Security in Distributed Data Warehousing .Available: http://www.ifs.univie.ac.at/~ww/sddw01.pdf.

E. Fernández-Medina, J. Trujillo, R. Villarroel, M. Piattini. (2006). Access control and audit model for the multidimensional modeling of data warehouses, *Decision Support Systems* 42 (3) pp.1270–1289.

E. Fernández-Medina, J. Trujillo, R. Villarroel, M. Piattini. (2007). Developing secure data warehouses with a UML extension, *Information Systems* 32 (6) pp.826–856.

J. Yang and J. Widom. (1998). Maintaining Temporal Views Over Non-temporal Information Sources for Data Warehousing, In proceedings of the 6th International Conference on Extending Database Technology, Advances in Database Technology – EDBT '98, pp.389-404.

J. Yang and J. Widom. (2001). Incremental computation and maintenance of temporal aggregates, In Proc. Int. Conf. on Data Engineering (ICDE '01), pp.262-283.

N. Rahman. (2008). Updating Data Warehouses with Temporal Data.AMCIS2008 Proceedings.Paper323 Available: http://aisel.aisnet.org/amcis2008/323.

Trujillo, J., & Luján-Mora, S. (2003). A UML based approach for modeling ETL processes in data warehouses. Springer *,Conceptual ModelingER. pp.*307-320.

P. Vassiliadis, A. Simitsis, P. Georgantas,M. Terrovitis, and S. Skiadopoulos. (2005). A Generic and Customizable Framework for the Design of ETL Scenarios. *Information Systems, Elsevier Science Ltd, vol. 30,* no. 7, pp. 492-525, Nov 2005.

P. Vassiliadis, A. Simitsis, S. Skiadopoulos. (2002). On the Logical Modeling of ETL Processes. *In Proc. of International Conference on Advanced Information Systems Engineering*, pp.782- 786.

T. Priebe and G. Pernul. (2000). Towards OLAP Security Design – Survey and Research issues. In Proceedings of the 3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00), pp 33–40, Washington DC, USA.

L. Wang, S. Jajodia and D. Wijesekera. (2004). Securing OLAP data cubes against privacy breaches**.** In Proceedings of the IEEE Symposium on Security and Privacy, pp 161–178, Berkeley, California.

R. Bruckner and A. Tjoa. (2002). Capturing Delays and Valid Times in Data Warehouses—Towards Timely Consistent Analyses, *Journal of Intelligent Information Systems,* 19, 2, pp.169 – 190.

R. Kimball, M. Ross and R. Merz. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*(2nd ed.),John Wiley & Son,(Chapter 4).

R. Villarroel, E. Fernández-Medina, M. Piattini and J. Trujillo. (2006). A UML 2.0/OCL extension for designing secure DWs, *Journal of Research and Practice in Information Technology* 38 (1) pp.31–43.

R. Villarroel, E. Fernández-Medina, M. Piattini, J. Trujillo. (2007). Developing secure data warehouses with a UML extension. Information Systems. Elsevier, Volume 32, Issue 6, September, pp.826-856.

Y. Yufei Tao, D. Papadias and C. Faloutsos. (2004). Approximate temporal aggregation. 20th International Conference on Data Engineering (ICDE'04), pp.190.

Y. Feng, H. Li, D. Agrawal and A. Abbadi. (2005). Exploiting Temporal Correlation in Temporal Data Warehouses, In proceedings of the 10th International Conference on Database Systems for Advanced Applications, DASFAA 2005,pp. 662-674.

Table 1. Special cases in TDW and their solutions

Case	**Solution**
Multiple versions of rows for a particular record, with the same primary key.	Each previous version will be expired with the row effective timestamp of the immediate next version of that row.
The same key has a DML code 'I' (insert) followed by 'U' (update)	Only the row with 'U' will be treated as the active row.
all rows with 'D' (delete) in the TransEvent table	Will be used to expire the corresponding row in the target table.
All the existing rows with 'U'	Will be expired in the target table and new rows inserted.
Delete non existing PK	The delete event is moved to invalid table

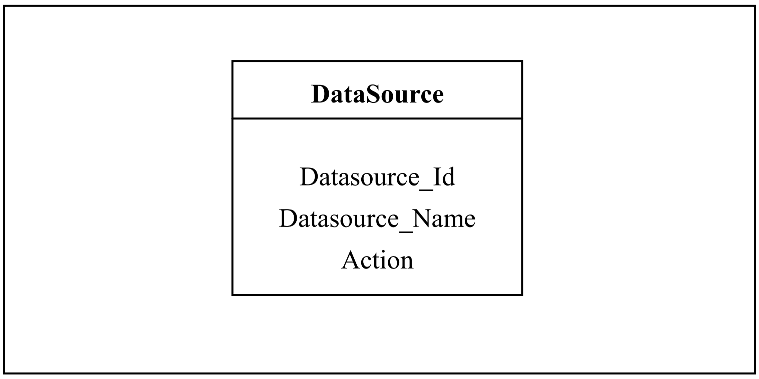

Figure 1. Data Source Class

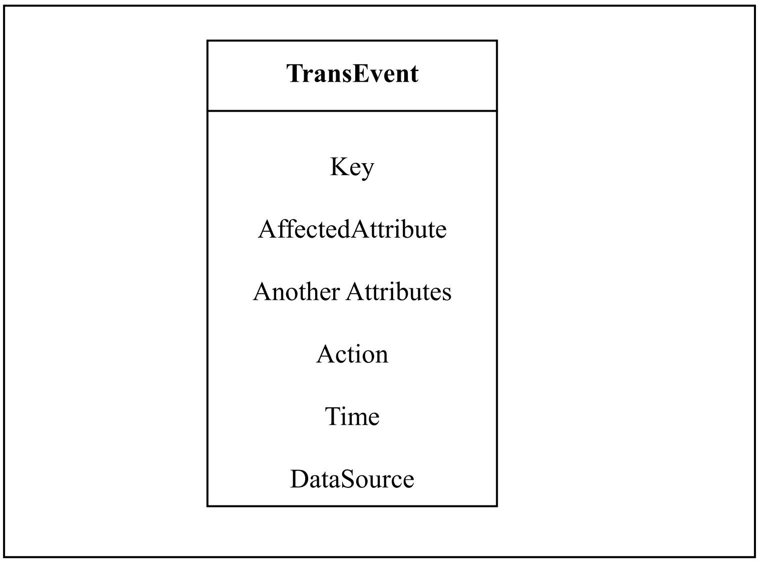

Figure 2. The TransEvent class

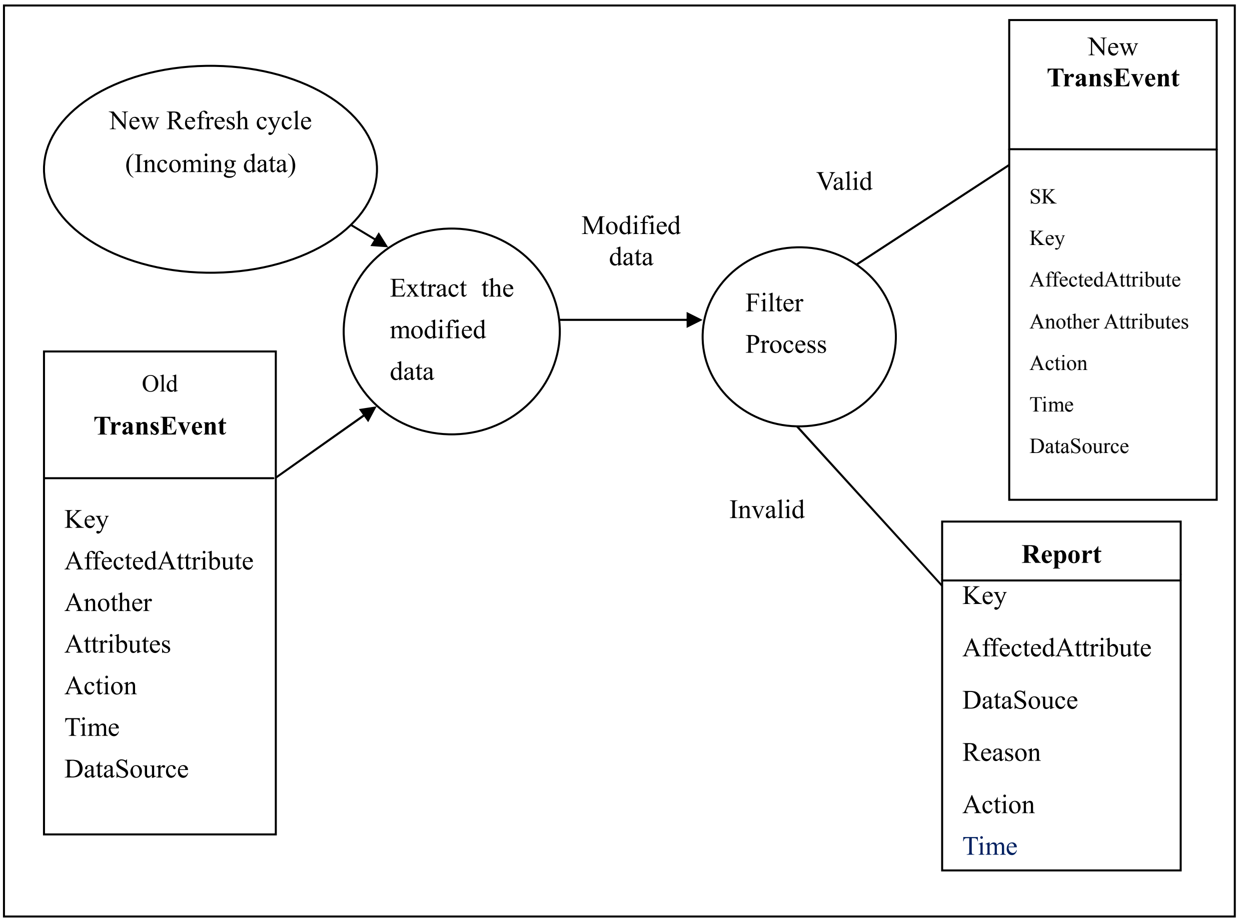

Figure 3. The Workflow of data Extraction in ETL

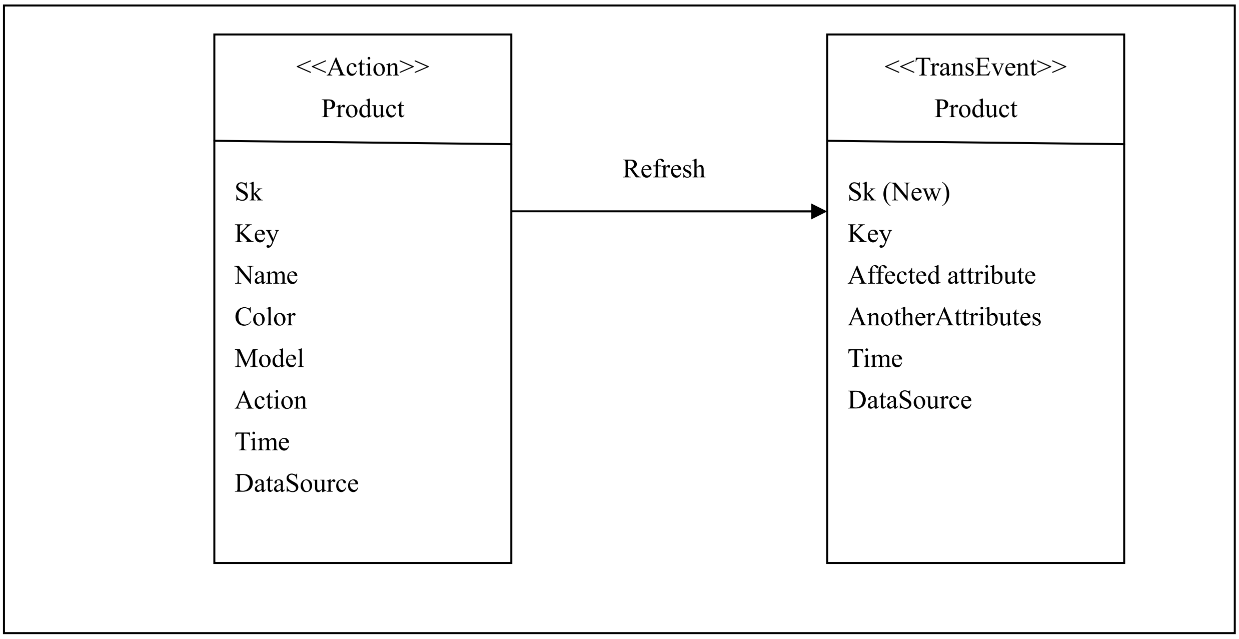

Figure 4. Example of SCD in ETL

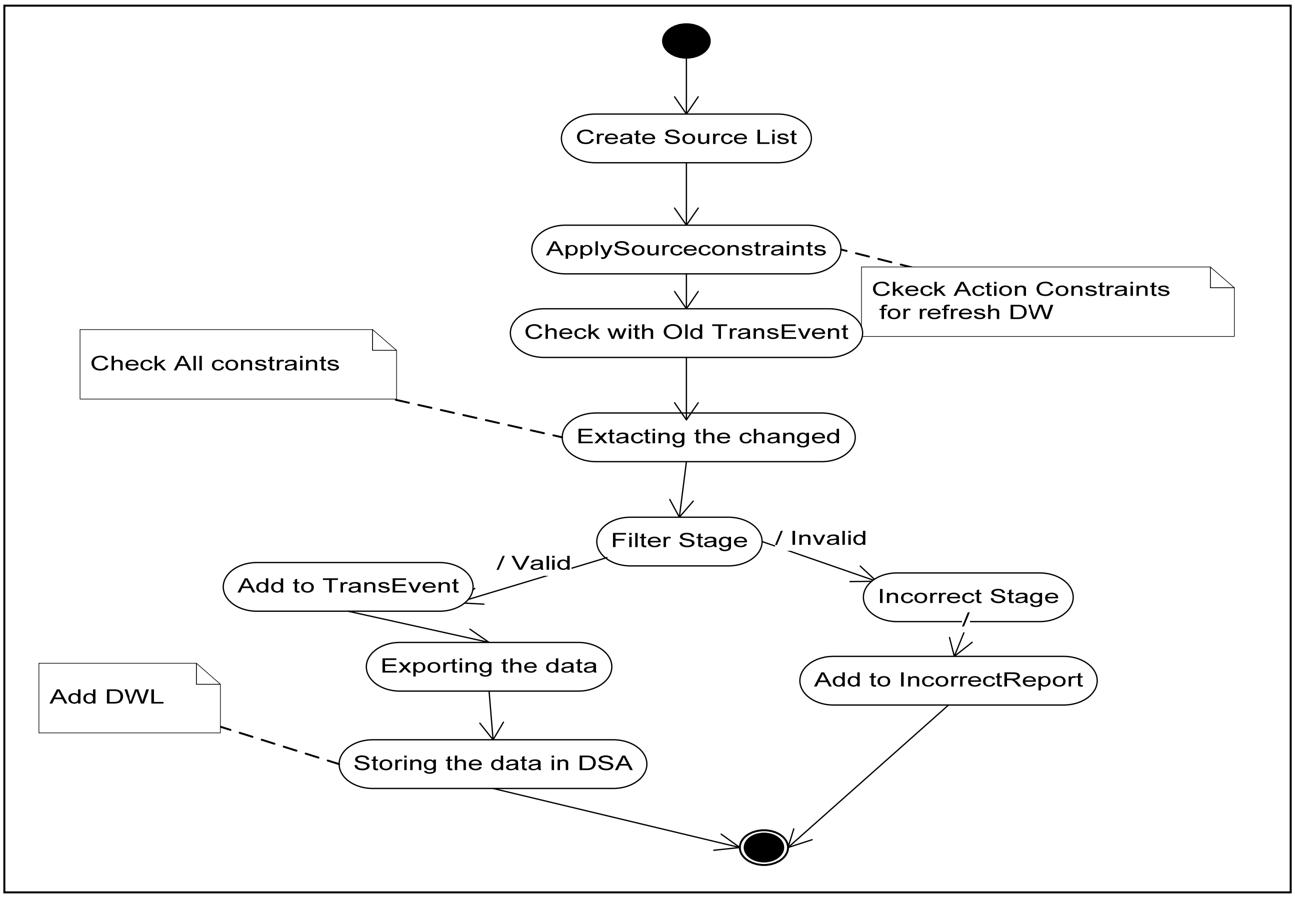

Figure 5. Activity Diagram of CMTDWS for ETL processes

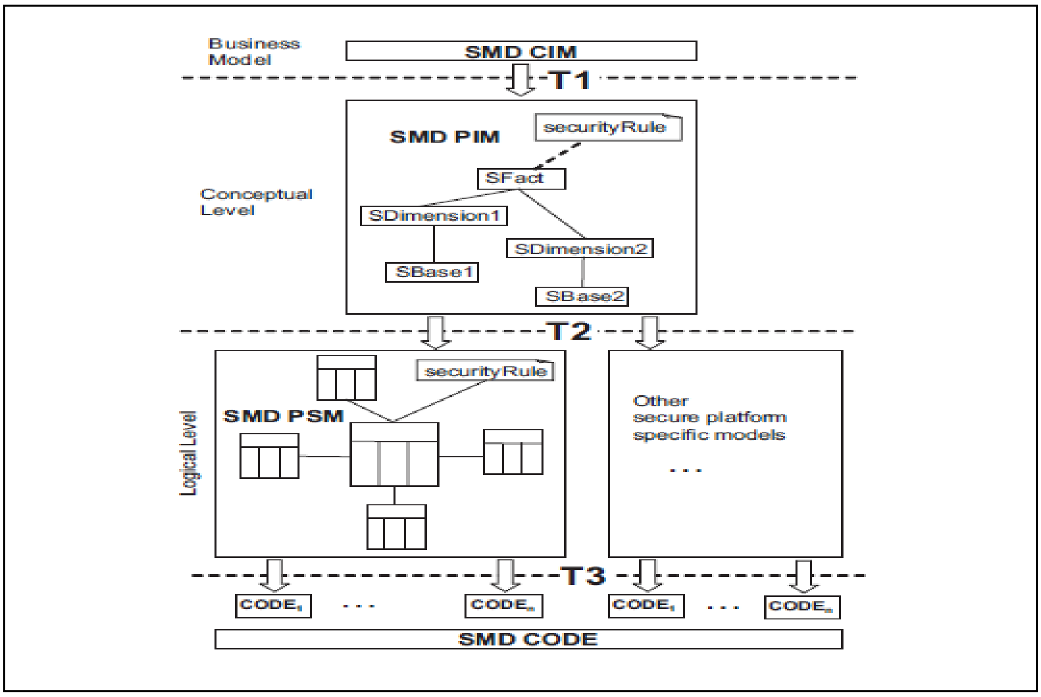

Figure 6. A framework for the development of secure DW

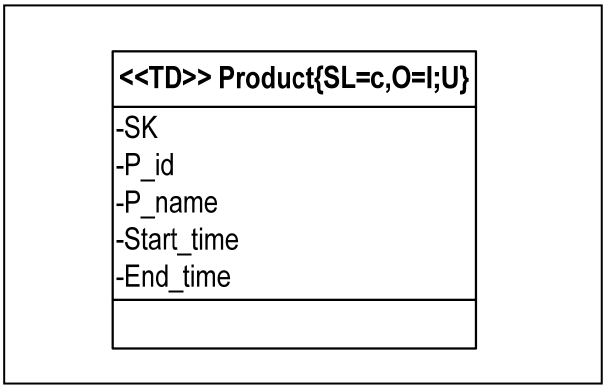

Figure 7. Slowly Changing Dimension (SCD) in DW

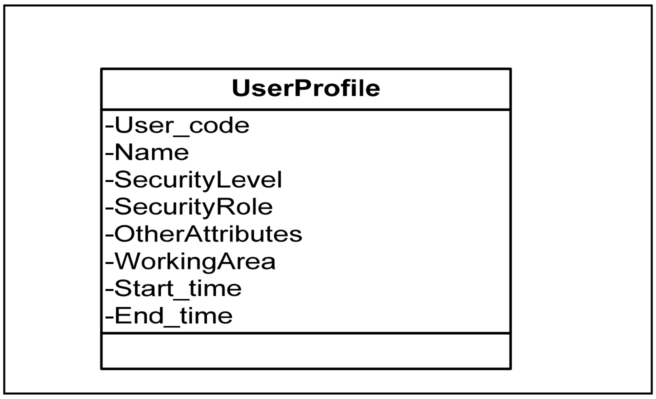

Figure 8. User profile class in DW