AN EXTENDED DATABASE REVERSE ENGINEERING – A KEY FOR DATABASE FORENSIC INVESTIGATION

P. D. Bodakhe[1], K. G. Bagde [2]

[1] Department of C.S.E., H.V.P.M.'s C.O.E.T. Amravati, Maharashtra, India, ***bodakhe.prajakta@gmail.com***
[2] Assistant Professor, Department of C.S.E., H.V.P.M.'s C.O.E.T. Amravati, Maharashtra, India

Abstract

The database forensic investigation plays an important role in the field of computer. The data stored in the database is generally stored in the form of tables. However, it is difficult to extract meaningful data without blueprints of database because the table inside the database has exceedingly complicated relation and the role of the table and field in the table are ambiguous. Proving a computer crime require very complicated processes which are based on digital evidence collection, forensic analysis and investigation process. Current database reverse engineering researches presume that the information regarding semantics of attributes, primary keys, and foreign keys in database tables is complete. However, this may not be the case. Because in a recent database reverse engineering effort to derive a data model from a table-based database system, we find the data content of many attributes are not related to their names at all. Hence database reverse engineering researches is used to extracts the information regarding semantics of attributes, primary keys, and foreign keys, different consistency constraints in database tables. In this paper, different database reverse engineering (DBRE) process such as table relationship analysis and entity relationship analysis are described .We can extracts an extended entity-relationship diagram from a table-based database with little descriptions for the fields in its tables and no description for keys. Also the analysis of the table relationship using database system catalogue, joins of tables, and design of the process extraction for examination of data is described. Data extraction methods will be used for the digital forensics, which more easily acquires digital evidences from databases using table relationship, entity relationship, different joins among the tables etc. By acquiring these techniques it will be possible for the database user to detect database tampering and dishonest manipulation of database.

Index Terms: *– Foreign key; Table Relationship; DB Forensic; DBRE;*

---***---

1. INTRODUCTION

Most company and organizations manage personnel, account, logistics etc through the database system. The business data stored these database is very important to investigate company dishonest act in the view of the digital forensics. But without blueprint of database it is not possible to obtain forensic evidences. Therefore, the investigator should confirm the contents of the database blueprints and check the relationship between the table and hidden table which unsigned blueprints. By using database reverse engineering methods it is possible to get data structure or blue print of database. The problems that arise in database reverse engineering fall in two categories that are addressed by the two major processes in DBRE, namely data structure extraction and data structure conceptualization. In addition, each of these processes grossly appears as the reverse of a standard database design process (resp. coding/physical design and logical design). The Data structure extraction process recovers the complete logical schema, including the explicit and implicit structures and properties. The Data structure conceptualization process recovers the conceptual schema underlying this logical schema[1].

In this paper, we will first derive a transformational model of DBRE, then we will describe the contents and the attributes extraction. Also we improve the database reverse engineering process and analyze the table relationship of database and data extraction method in the view of the digital forensics. Moreover we proposed the efficient tool which more easily acquires digital evidences from databases using table relationship and data extraction.

Current database reverse engineering researches presume that the information regarding semantics of attributes, primary keys, and foreign keys in database tables is complete. However, this may not be the case. In a recent DBRE effort to derive a data model from a table-based database system, we find the data content of many attributes are not related to their names at all. Thus, in this paper, along with table relationship we present a process that extracts an extended entity-relationship diagram from a table-based database with little descriptions for the fields in its tables and no description for keys[2],[6].

2. EXTRACTION METHOD FOR ENTITY RELATIONSHIP DIAGRAM

Before starting the data structure extraction we have to collect some information related to database. In the step of projection preparation, we utilize screen displays to construct form instances. To automate most of the process, we transplant the original legacy data into a relational database system, SQL server, so that SQL commands may be applied to analyze the data. Secondly, code analysis and data analysis involving comparisons of fields and decomposition of fields are applied to extract attribute semantics from forms and table schemas, followed by the determination of primary keys, foreign keys and constraints of the database system. In the final step of conceptualization, with the processes of table mergence and relationship identification, we derive the first-cut entity-relationship schema. Finally, we apply some domain knowledge to construct the complete conceptual structure. The first step of DBRE is the preparation of system profiles for the DBRE project. Before starting the DBRE work, we have to collect and organize the complete description of all data assets in the database. The description could be divided into three types:

1. Explicit data description language (DDL): for instance, the name, domain and length of fields in database tables.
2. Implicit structure: for instance, primary keys, foreign keys, constraints, and program process logic for accessing the database.
3. Other parts in programs: such as constraints on data structures in application programs.[5],[6]

In data structure extraction we extracts the complete database schema. If there is a formal DDL description of the database, this process could be greatly expedited. Otherwise, a fair amount of data analysis, program analysis, and form analysis need to be performed. Data analysis examines the content of a database to uncover properties such as uniqueness and test hypothesis such as foreign keys. The users' domain knowledge and judgment are the decisive elements for semantics. Program analysis provides extra information in integrity constraints. There are three steps, attributes extraction, keys extraction, and constraints extraction in this extraction process. The aim of attributes extraction is to extract semantic information for database fields through field comparison, character comparison, data analysis, and code analysis[3].

2.1. Attribute Extraction

Field comparison is to compare form fields and database fields through instances in order to obtain the true meaning of each database field. This is necessary because the meanings of some of database fields could not be referred from the title name. First, we find the corresponding entries of forms fields in database fields, and then use captions of form fields and the context of the form to derive the meaning of database fields. In the process of comparison, form instances are the medium. Since the value of each field in form instances is different, by comparing values in form fields and database fields, we could identify the corresponding database fields readily. In this way, we are able to extract the semantics of most attributes in the system.

The rule is formulated as follows:

Let an unique value, v, is inputted in a display form under the caption, c, and there exists a table, t, which contains under a field, f, that is, v 2 Pf(t), where P is the project operation of the relational algebra, then the meaning of the field f is the same as the meaning of the caption c. The values of Field Schema Data and Attribute Test Value are compared. If the values of the two fields are the same, it means that the two fields match, and the Attribute. Attribute Name should stand for Field-Schema. After the process of field comparison, there are still some form fields with no corresponding database fields. After further investigation, we find that this is due to the fact that certain database fields contain data from several form fields. The reason for packing different sources of input data into one field is possibly the result of a quick and dirty solution to accommodate additional data items by stuffing these data into fields with spare space and avoiding the trouble of restructuring the table schema. Such packing is possible only when the data type of the field is fixed-sized character strings. The database field is, as a result, overloaded with more than one meaning which may not be related in the sense of a composite attribute in the ERD design. That is, there are actually several attributes inside such field.

The previous rule can be extended to handle such overloaded fields as follows:

Let an unique string, s, is inputted in a display form under the caption, c, and there exists a table, t, which contains s as a substring under a field, f, that is, $v 2 Pf (t) such that s is a substring of v, then the field f is a overloaded field and the meaning of the field f contains the meaning of the caption c. Therefore, besides simple field comparison, we need to perform character comparison that employs characters searching to find the corresponding database fields for these form fields. After the process of field comparison and character comparison, the semantics of the database fields that can be directly related to captions in form fields through unique form instances are extracted. For those attributes that does not store unique form instances, data analysis and code analysis are applied manually to extract other attributes from database fields. Data analysis is based on the existing content of the database to classify the values of fields, and then applying domain knowledge to identify the differences between the classified catalogues. By comparing the differences among catalogues, combined with domain knowledge, we might be able to understand the meaning of the field. If there is no sufficient knowledge to resolve the meaning, some code analysis will be needed. Code analysis procedure usually searches source code for the field name in question, analyzes the related processing of the field and then resolves the meaning of the field. Such analysis can be augmented with form fields for a database filed may be related to a form field after some computation. Some database fields that could not be observed in forms are produced by programs automatically without referring to any other fields. Such fields are considered not of any semantic importance in this case study, and discarded in subsequent analyzes[4][6].

2.2. Key Extraction

The second step of data structure extraction is key extraction to obtain primary keys and foreign keys for the database tables. With the knowledge of primary keys and foreign keys, the referential association among tables can be constructed. Primary keys and foreign keys are declared in a relational database system, but such information is not explicit for dBase systems. The dBase system usually maintains an index file to speed up data searching for a table. Usually, primary keys are included in such index file; therefore, it is a good starting point to extract primary keys. If the row count of an index file is the same as that of the indexed file, the index file could be selected as candidate keys. If there are more than one candidate keys, the one containing less fields would be chosen as the primary key. For example, let us consider that there is a table StDat Now with three indexes. The first one is St_Id, the second one is St_Dyc and the third one is St_Id+St_Dyc.

Table -1: Field comparison example

Database FieldSchema			Form Attribute	
TableName	FieldName	Data	AttributeName	TestValue
StDatNow	St_Id	11111111	Student number	11111111
StDatNow	St_Pass	*MA2222	Password	2222
StDatNow	St_Dyc	ch01101	Subject, grade, class	ch01101

FieldSchema Document		
TableName	FieldName	Meaning
StDatNow	St_Id	Student Number
StDatNow	St_Pass	* (not find)
StDatNow	St_Dyc	Subject,grade, class

The row count of the table is calculated with a SQL count command to be 4704. And then, the row counts of the three indexes with distinct values are also calculated to be 4704, 158, and 4704 for St_Id, St_Dyc, and St_Id+St_Dyc, respectively. Therefore, St_Id and St_Id+St_Dyc are candidate keys, and St_Id is chosen as primary key for containing fewer fields. As for tables without an index file, we have to examine all the values for each fields in the table for uniqueness. However, the fields containing null values can be eliminated to reduce the effort of checking uniqueness. Finally, the same criterion is followed to pick out a primary key. After primary keys are established, we can apply primary keys to extract foreign keys in order to identify association among tables. During this process, every primary key is checked whether it is referred in fields of other tables. The referring field is a foreign key. The criterion is that the domains of the referring fields and the referred fields must match and the values of the referring field must be a subset of those of the referred field. Such criteria can be easily tested by designing appropriate SQL commands[3][7].

2.3. Constraint Extraction

The primary objective of constraints extraction is to obtain the association cardinality between primary keys and foreign keys. If a value of the primary key in a table shows in only one record in another table with the associated foreign key, the mapping cardinality is inferred to be one to one. Otherwise the mapping cardinality is considered one to many. In fact, if the foreign key relationship is already established, one only needs to check for the uniqueness of values in the foreign key fields to determine the cardinality. If the result of the query is not zero, there exists at least one occurrence in TA with its primary key value equivalent to multiple foreign key values in TB. Therefore, it is a one to many mapping. Otherwise, it is a one to one mapping. Though such procedure serves most of the purpose, cardinality constraints on relationships involving more than two tables should be determined by the algorithm. For example, we apply this procedure to calculate the cardinalities between the primary key St_Id of table StDat-Now and a foreign key, St_Id of an other table say StNua. With the result of the query not zero, we determine that the mapping between StDatNow.St_Id and StNua.St_Id is one to many. At the end of this process, a logical schema of the database comes into shape[6][8].

3. EXTRACTION METHODS FOR DATABASE TABLE RELATIONSHIP

A. Method based on the Application Software for analysis of table relationship

The application software such as business program accesses the database and automatically performs the task instead of user. The application software execution database works such as write, modify, and delete etc by creating and sending database query or calling the procedure stored in database. The method based on the application consists of two parts; monitoring and reverse engineering. The Monitoring is that the application software captures query packet and analyzes Inner Join and Outer Join of the query, and then extracts relationship. The reverse engineering extracts relationship through source code which generate query inside application or database procedure and parameter. This method search table relationship by analyzing those data and matching them with investigation data. However, method based on the application software for analysis of table relationship requires same database and application software and investigator need enough time to analyze. Furthermore, since application software and database are designed uniquely for each company and institutions, it is difficult to analyze depend on software complexity. And according to the investigators of the reverse engineering capability, it determines whether the analysis is possible or not.[3]

B. Method based on the database for analysis of table relationship

Common DBMS offers DBMS system catalog information about database as to view format. Based on the system catalogue information, we can analyze relationship between the tables and these relations confirm by comparing with application software outputs . The database table consists of several field, these field has attribution which shows relationship with other table. In case of the foreign key, we can verify the relationship between the tables as using key in the other table. Moreover, the tables which include relationship have same field data type and length; we should compare these two tables to confirm the relationship There exist restrictions in database table and these restrictions are applied equally. Therefore, we should check the conditions field by field and compare the conditions to confirm table relationship. As mentioned above, we have to check the relationship by comparing with reference table which is target to analyze such as key, data type, length, and condition etc. the relationship has 1 vs. n, 1 vs. 1, n vs. n type, we confirm this relationship through the cardinality search. Unlike analysis of application software, database method can be applied other database. Therefore we can design the automation tool for various databases. However, method based on the prior automated database for analysis of table relationship cannot extract exactly same table relationship as provided by application software[3][9].

4. DATABASE CONCEPTUALIZATION

The goal of this phase is to recover the complete DMS schema, including all the implicit and explicit structures and constraints. As explained above, the main problem of the Data Structure Extraction phase is to discover and to make explicit, through the Refinement process, the structures and constraints that were either implicitly implemented or merely discarded during the development process. We can define the concept of implicit construct, or describe the DDL code extraction process to analyze the problems and construct elicitation techniques of implicit.

CONCLUSIONS

In this paper, we have presented a DBRE approach that supports extracting an extended entity-relationship diagram from database table. Also we analyze the table relationship using database system catalogue, design the extraction process for examination of data using DBRE. At present, there exist only the tools with database design view and query execution tool. However, for digital forensics, there is not the tool which is possible to search, join, extraction the table relationship. Therefore, since DBRE is easy to use without any professional database knowledge, it can make digital Forensic investigation much more convenient.

REFERENCES

[1] Kyriacos Pavlou and Richard T. Snodgrass,“Forensic Analysis of Database Tampering” SIGMOD’06, June 27–29, 2006, Chicago, Illinois, USA.
[2] Jasmin Azemovic and Denis Music , “Methods For Efficient Digital Evidences Collecting Of Business Proceses And Users Activity In eLearning Enviroments” 978-0-7695-3948-5/10 $26.00 © 2010 IEEE DOI 10.1109/IC4E.2010.92
[3] Dongchan Lee, Jaemin Choi, Sangjin Lee, “Database Forensic Investigation based on Table Relationship Analysis Techniques” 978-1-4244-4946-0/09/$25.00 ©2009 IEEE
[4] Peter Frühwirt, Markus Huber, Martin Mulazzani, Edgar R. Weippl, “InnoDB Database Forensics” 1550-445X/10 $26.00 © 2010 IEEE DOI 10.1109/AINA.2010.152
[5] Kyriacos E. Pavlou and Richard T. Snodgrass, ” The Tiled Bitmap Forensic Analysis Algorithm”
[6] Dowming Yeh, Yuwen Li, William Chu, Extracting entity-relationship diagram from a table-based legacy database. in proc. The Journal of Systems and Software 81, Elsevier Science Ltd, 764–771, 26 July 2007.
[7] “SQL Server Database Forensic” USA 2007
[8] Patrick Stahlberg, Gerome Miklau, and Brian Neil Levine, “Threats to Privacy in the Forensic Analysis of Database Systems” SIGMOD’07, June 12–14, 2007, Beijing, China.Copyright 2007
[9] Alhajj, R., Extracting the extended entity-relationship model from a legacy relational database. in proc. Information Systems 28, Elsevier Science Ltd, 597–618, 29 May 2002