Research Journal of Recent Sciences ISSN 2277-2502
Vol. **2(5),** 49-58, May **(2013)** *Res. J. Recent Sci.*

Database Reverse Engineering Methods: What is Missing?

Natash Ali Mian[1], Sher Afzal Khan[2] and Nazir Ahmad Zafar[3]
[1]Faculty of Information Technology, University of Central Punjab, Lahore, PAKISTAN
[2]Department of Computer Science, Abdul Wali Khan University Mardan, PAKISTAN
[3]Department of Computer Science, King Faisal University, Hofuf, SAUDI ARABIA

Available online at: www.isca.in
Received 14th December 2012, revised 28th January 2013, accepted 28th February 2013

Abstract

Database reverse engineering (DBRE) is a process of extracting requirements from a running system. The process is carried to understand the system having least design and architecture documentation. DBRE is very beneficial when we are expanding our systems or shifting to a newer technology. Since its start, reverse engineering has given hard time to software engineers but researchers are doing a continuous effort to make the process more effective and efficient. Many frameworks and methods have been proposed in the literature; each has its inputs, limitations, assumptions and outputs. CASE tools have been developed by researchers and commercial companies, but no tool provides a complete set of features to reverse engineer a database to a conceptual schema, also a concrete formal model is missing. Few common problems arise while reverse engineering of any system using existing approaches. In this review paper we have analyzed different methods proposed in the literature, discussed their strengths and limitations. Major problems faced by the reverse engineers while using existing methods are identified and discussed. This paper will motivate the software engineers to develop a DBRE method which takes minimal set of inputs, applies least assumptions and generates better output with least or no human intervention.

Keywords: Analysis of DBRE methods, Automated reverse engineering, Database reverse engineering

Introduction

DBRE is a procedure of digging out requirements from a system in operation. Usually, legacy systems have least design documentation and thus make the maintenance job difficult. Improvement of such software systems can be done with significantly reduced amount of effort and cost if the conceptual models of these systems are readily available. While reverse engineering, a reverse engineer takes previous implementation that embodies a design and retrieves essential problem domain content, discards design optimizations and implementation decisions. DBRE has a very important role in this regard and has been a topic of research since 1980s.

There are different types of conceptual models used in designing software applications, for instance, Unified Modeling Language (UML), Object Modeling Technique (OMT) and Entity-Relationship (ER) model. ER model is widely used as a data modeling tool for database applications due to its ease of use and representation. An Enhance Entity-Relationship (EER) model is an extension of ER model which has additional constructs such as specialization (or generalization), shared subclasses, union types/ categories. It should be noted that the constructs of an ER (or EER) model are used for comparison in this paper. The constructs in OMT can be easily mapped to those of the EER model. Similarly ERC+ model is included in EER model columns for ease of reference during analysis. Constructs of ER model and the additional constructs of EER model are- regular entity type, weak entity type, simple attribute, composite attribute, key attribute, multi-valued attribute, relationship type, structural constraints, specialization, specialization constraints, shared subclass and union type.

We need reverse engineering to improve existing systems as software evolves with the passage of time which is due to continuous change in requirements of industry. Hence we require a good understanding of implemented system so we can extend the system in a better and efficient way. One of the problems while reverse engineering is lack of documentation by developers. We find minimum design details of implemented system. Also each development team follows its own standard and criterion to develop a system; this makes the job of a reverse engineer even more difficult. Every time we are reverse engineering a system we come across new problems.

The objective of this research is to analyze the capabilities of existing methods to reverse engineer an existing system. For this purpose, we have studied eleven different DBRE methods and provided a comparison which elaborates their strengths and limitations. It is to be noted that we have applied the existing techniques to small examples to better understand the working of each method and analyze the outputs produced by each method discussed in this paper. We believe that this research will motivate the researchers to produce a method which can extract a model representing the existing database with least or almost no human interaction. It is to be noted that initial results of our work have been presented in our earlier paper[1].

This paper is organized in four sections –the background of this research and existing work in the field of DBRE, next section explores the capabilities of each method studied, analysis and results section provides a comparative analysis of their capabilities and finally we conclude this research and provide pointers to future work.

Related Work: In literature[2-10], we find many DBRE techniques which can convert a given database to a high-level conceptual model. In some previous papers a good survey and comparison[10,11] of existing techniques is given, whereas issues involved in DBRE as well as some of the solutions are discussed[7]. There has been a lot of work in area of reverse engineering during the last two decades. Few tools have been developed[12,13] and some methods have been proposed, however a complete automated tool which extracts all features of conceptual model is still missing. Researchers have used multiple ways to extract information from tables, data, code, data dictionary and forms[13-23]. All have contributed to the knowledge of reverse engineering.

A survey of existing methods is given with their strengths and limitations[17]. Few problems have also been identified by the authors. Mens and Tourwe have used formal concept analysis to reverse engineer code[14]. Ichise and Ishida have used cellular automata for reverse engineering of spatial patterns[15]. Cabot et al. have applied their model to reverse engineer constructs from an object relational database schema[24] and generated UML diagrams[16]. Lammari et al. have proposed an approach to extract generalization hierarchies from a relational database and have given steps of refinement which increase the quality of the output produced[18]. Mammar and Laleau have given a formal approach for the forward engineering of database systems[19]; they follow a step wise approach and take three steps to model a database system. Their approach starts by designing a UML diagram which is transformed in to B specifications and finally this specification is converted in to SQL statements for implementation. Abbasifard et al have given a method to integrate the databases developed in older systems with the new ones and they have used reverse engineering to extract the information from the existing systems[20]. Authors propose a model to integrate the existing databases in to modern systems, their focus is on primary keys, foreign keys and normalization[25]. Masoud et al. have given a simple yet effective method to reverse engineer an existing database and have developed a CASE tool[21]. Ghalayini et al have used domain ontology to extract the domain conceptual model and have proposed ways to integrate multiple data models and information resources[22]. Astova and Stantic have proposed an approach to extract conceptual model by analyzing HTML forms, they are of the view that forms give sufficient information to deduce semantics of a relational model[13]. They use ontologies in the process of extraction of the conceptual model. Alhajj has proposed a method of extracting an extended entity relationship model from a legacy relational database, still the method does not extracts all constructs of an extended entity relationship model[23]. It is to be noted that this method is an extension of his previous model proposed earlier in 2001. Benslimane et al. propose a reverse engineering approach using domain ontology for semi structured web applications which do not have documentation[25], ontology has been successfully used for web as well[26]. The proposed approach is limited and does not extract all concepts from the application. Trinkunas and Vasilecas propose an approach in which they develop ontologies from the given relational database[27]. Their approach follows graph oriented model for conceptual data model. In the given approach a conceptual model is developed from relational database[28, 29] using reverse engineering. Ontologies are extracted using CASE tool OntER automatically, all transformations are formally defined and can be reused.

Authors propose a technique to discover conceptual schema using association mining[30], a data mining technique with success[31]. The discovered schema is at third normal form. An algorithm for finding the minimal branching dependencies between a given set of attributes and a given attribute in a relation of a database is proposed, this is very useful contribution in the overall process of DBRE[32]. Authors introduce a concept of annotation based query answer to address the inconsistency of data; focus of their work was on functional dependencies[33]. A process to extract cardinalities in a given database that takes data dictionary and data as input and generates results by analysis of mentioned inputs[34]. Anthony Cleve and colleagues propose an approach to perform the task of DBRE by data reverse engineering where they construct system dependency graphs (SDGs). Their approach successfully computes the program slices of a database system[35].

Authors have proposed a new method to reverse engineering object relational database which is based on the execution of a set of transformation rules and generate a UML based conceptual schema that represents the current database at a higher level of abstraction[16]. An industrial example to reverse engineer an existing database schema is proposed[36]; technique that is followed is discussed previous papers by same authors[37, 38] whereas the dimensions of DBRE have been summarized in later papers[39]. Authors present a case study while reverse engineering an industrial sub system, problem faced and lessons learned have also been discussed in the paper[40].

Authors propose a transformational reverse engineering approach and claims that quality of output is improved by using this methodology[41].

Authors propose criterion for checking the completeness and accuracy of the reverse engineering process[42]. Chu et al. have applied transition net, a formal approach to reverse engineer software components and the results are annotated by the use of predicates[43]. They have used a variety of petri-nets which help in revealing the information about the structure and dynamic behavior of the modeled system. A model to evaluate the quality of reverse engineering process is presented[44]. Authors apply fuzzy logic and formal methods to reverse engineer a given database[45]; they use petri nets to perform the task. An incremental approach to reverse engineering of existing systems is given[46]. They apply a divide and conquer approach to convert a system. A source transformation technique to bridge the gap between data modeling and application modeling is discussed[47]. They have also proposed a tool which can recover a UML based ER diagram from a SQL data definition language schema. A reverse engineering approach for web data is given[48]. An approach for extraction of ontology form from relational databases is proposed[49]. Authors have proposed an approach for reverse engineering of existing database schemas into conceptual schemas with minimum inputs[50]. Perez et al. give a methodology of reverse engineering using formal method[51]. Angyal et al. give an overview of the state of art technology in area of reverse engineering[52]. A detailed comparison of approaches and tools is also given in the paper and major emphasis of authors is on code reverse engineering. Different strategies of data reverse engineering are summarized and an approach to extract a conceptual schema from given data is presented[53]. A roadmap of reverse engineering which emphasizes on the importance of the field in coming times is given, it also emphasis on including it as a part of curriculum[54]. The importance of incorporating reverse engineering concepts and techniques into software engineering curriculum is highlighted[55]. A broad view of reverse engineering discipline is presented and authors try to summaries the work done in last two decades, it also gives directions of research in the field of reverse engineering[56]. Canfora and Penta give an overview of the work done in the area of reverse engineering[56]. Multiple studies[57-60] show that formal methods can be applied to variety of software systems with a very high success rate which motivates us to apply formal methods to DBRE process.

It is important to note that these methods are restricted by large input requirements, assumptions and they require human interference in the process of reverse engineering. Each method has its own inputs and outputs under certain limitations. Data instances, relations-schemes, primary key (PK), inclusion dependencies (IND), functional dependencies, code or display forms, ER, EER or OMT model are few examples of inputs and outputs whereas third normal form (3NF), boye codd normal form (BCNF), consistent naming conventions are examples of assumptions and limitations. However most of the methods require human intervention. Due to this different software engineers may reverse engineer a system in multiple ways according to their expertise and experience. Hence human intelligence also reduces the consistency of results. This is the aspect of reverse engineering where the practitioners do not have an automated process to completely reverse engineer a system. Hence, we require a formal model with well defined inputs, processes and outputs; this will not only define a strong mathematical model, but also minimize the human intervention, therefore an automated model using formal methods is required which will lead to automated reverse engineering.

Keeping in view the importance and need of DBRE, number of CASE tools have been developed in research[12] as prototypes whereas some are commercially available[61]. These tools differ in what they need for input and what they produce. Some tools require just relational schema whereas others need data instances as well. Some tools additionally require application code, and/or forms to start DBRE process. Some tools assume that database is normalized up to BCNF whereas others can start even from 3NF. Likewise, for output, it can be an ER, EER or an OMT model[61].

In this paper we have selected eleven DBRE methods. There are many other techniques in literature which may be used in certain conditions or may be used to carry some steps of DBRE, however, none of them gives a complete model to reverse engineer a given database schema. Only those methods are considered for analyses which perform the complete process of database reverse engineering.

Reverse Engineering Methods

Many methods exist in the literature, each having its own inputs, outputs and assumptions. We will briefly discuss each method in this section.

Alhajj's Method: A method to reverse engineer relational database schema with least information[2]. It takes a legacy database as input and generates an ER model. The method has four steps to perform DBRE. This method gives a simpler way to discover a conceptual schema from a conventional database. Determination of cardinalities is a major contribution of this method. The method takes no assumptions; hence, practical applicability of method increases to wide number of systems. However, the method expects user guidance in semantic decisions which otherwise cannot be inferred from the conventional database schema.

Andersson's Method: Author has proposed a method to extract an ERC+ model from data definition language (DDL) and data manipulation language (DML) statements[3]. ERC+ model is an extension of ER model with some additional constructs (multi-valued and complex objects). It takes application code, DDL and DML statements as inputs. It performs the reverse engineering task is five steps. It is to be noted that human interference is required to apply the transformation rules to initial schema generated. Method is applied to a real time system to verify its practical applicability and results obtained were encouraging.

Navathe and Awong's Method: Navathe and Awong's method requires relation schemes assumed to be in 3NF or BCNF as input[4]. Other assumptions are - names of attributes follow consistent naming conventions, all possible candidate keys (CK) of given schema are specified, there must not be any ambiguity in foreign keys (FK) and no homonyms must be present in the schema. The output is an EER model. This method takes three steps to reverse engineer a relational schema. The method emphasizes on most common situations that arise in practical problems and does not claim exhaustiveness. Though this method has a very small set of inputs; however its assumptions reduce its applicability to large set of systems.

Chiang et al.'s Method: Chiang et al.'s Method requires data instances, relation-schemes, PK and inclusion dependencies (Optional) as inputs[5,6]. Relations are required to be in 3NF, names of attributes are consistent; following some convention and values of key attribute have no error. EER model is generated as an output. Major contribution of this method is to find inclusion dependencies with justification of the transformations applied. The method takes nine steps to reverse engineer the given inputs. It is to be noted that steps and cases where human input is required are identified in the method. Method requires human intervention which reduces consistency of output.

Johannesson's Method: This method requires FD's and IND's as inputs and relation schema is assumed to be in 3NF[8]. The output is a conceptual schema described as a pair containing a language L and a set IC of typing, mapping and generalization constraints. The method has four major steps. One of the reasons this method is limited to a small set of problems is that it is very demanding on assumptions; it requires relations in 3NF, availability of IND. This information is not available for many of the old databases; this makes its use limited to only those systems which give this information.

Yeh and Li Method: Athors have proposed a method to extract an EER diagram from a table based legacy system[9]. Method requires display forms and table schema as inputs; output is an EER model. Authors have given a case study to understand the process they have devised. The approach uses display forms, table schema and instances as inputs and takes six steps to perform the reverse engineering task. This method requires many iterations so process becomes slower; forms may also give misleading results because they do not reflect the exact picture of database working behind.

Premerlani and Blaha's Method: Premerlani and Blaha's Method requires relation schemes and data as inputs with no mandatory assumptions. It is to be noted that the method produces an object modeling technique (OMT) model rather than an ER/ EER schema; however the transformations performed are very similar to those done in case of ER/ EER model[38]. This method takes seven steps to recover the OMT model. Transformations are performed by following the guidelines in section six of their paper[38]. Lot of human intervention is required to carry the process which effects the output. Also the reverse engineer should have a high skill level when using this method.

Markowitz et al's Method: This method takes relation schemes, key dependencies and key based IND's as inputs; relations are assumed in BCNF and output is an EER model[62]. The method takes four steps to convert schema to an EER model. The method is very demanding on input side and all these are not available in practice for most of the systems, this makes this method less usable to practical problems.

Table-1
Analysis of Inputs, Outputs and Assumptions of different Reverse Engineering Methods

Method	Inputs	Assumptions	Outputs
Alhajj[2]	Conventional database schema	-	ER Model
Andersson[3]	DDL, DML, code	-	ERC+ Model
Navathe and Awong[4]	Relational Schemes	3NF or BCNF, no ambiguities in Foreign Key's, no homonyms, all Candidate Key's available.	EER Model
Chiang et al.[5,6]	Data, Relational Schemes, Primary Key, Inclusion Dependency	3NF, consistent naming of attributes and No Error on key attributes values	EER Model
Johannesson[8]	Relational Schemes, Functional Dependency, ND	3NF relations	ER Model
Yeh and Li[9]	Display forms, table schema	Grouping of fields in forms represents true picture of Database.	EER Model
Premerlani and Blaha[38]	Relational Schemes, Data	-	OMT Model
Markowitz et al.[62]	Relational Schemes, Primary Key, Inclusion Dependency	BCNF relations	EER Model.
Petit et al.[63]	Relational Database, Data and code	with unique and not null constraints,	EER Model
Signore et al.[64]	Relational Schemes, Code	-	ER Model
Mfoura[65]	Set of forms, database schema	Forms represent exact picture of Database schema	ER Model

Petit et al's Method: Petit et al's method has no mandatory assumptions and requires relation schemes with unique and not null constraints. Data and code is required to re-engineer the relational model. The output is an EER model. This method takes four steps to process the inputs. This method requires high human intervention in all steps, user interaction is also required to carry the process efficiently; these effect the consistency in quality of output[63].

Signore et al.'s Method: Signore et al.'s Method has relation schemes and code as inputs with no assumptions to produce an ER model[64]. The method takes three major steps to generate an ER model. One of the limitations of this method is that it requires high human intervention. Also code analysis is required to get the best out of the process.

Mfoura's Method: Author has proposed a methodology to reverse engineer a relational database from a set of form model schema[65]. Inputs are data input forms and output is an ER schema. Form model schema gather information, structural information and constraints among data. This information is extracted from both forms structures and instances after generalizing them into database semantics using a guided inference process. This method requires human intelligence which affects the quality of results produced. Also skill level of the software engineer affects the quality of results produced.

Analysis and Results

This section gives a comparative analysis of eleven reverse engineering methods studied.

Analysis of Existing Methods: Table-1 gives a summarized view of inputs, outputs and assumptions of each method, whereas table-2 gives strengths and limitations of each method. It is to be noted that most of methods listed require human intervention to get required output. Despite of strong theory for mapping of an ER to a relational model we cannot reverse engineer a relational model completely. This is due to the variability of problems while reverse engineering. Also most of methods take large set of assumptions which does not reflect the true picture of real world.

Table-2
Strengths and Limitations of Reverse Engineering Methods

Method	Strengths	Limitations
Alhajj[2]	generates all possible Candidate and Foreign Keys, determines Primary Key, extracts cardinalities from.Takes no assumptions for Input	requires Human Intervention in few steps which otherwise could not be completed
Andersson[3]	generate an ERC+ model which includes weak entities, complex objects & cardinalities. Takes no assumptions	requires lot of Human Intervention to perform. Lot of processing is required
Navathe and Awong[4]	simple in understanding with minimum Input requirements to produce a Conceptual Schema	large set of assumptions & Human Intervention, problems handled on case basis
Chiang et al.[5,6]	generates Inclusion Dependency , complete justification of conversions applied to the existing database. It can be widely applied to practical problems and can generate an EER schema	lot of Human Intervention is required. For new type of system domain knowledge is required.
Johannesson[8]	based on concepts of Relational Database theory. Defines clear steps to Reverse Engineer Relational Database schema. Identifies steps to Reverse Engineer Database in an automatic manner	Method needs all keys & Inclusion Dependencies which are generally not available
Yeh and Li[9]	identifies key attributes from given legacy system, uses a system with least information of keys	requires lot of processing
Premerlani and Blaha[38]	defines very small steps to Reverse Engineer a system and takes no mandatory assumptions	Almost all steps require Human Intervention
Markowitz et al.[62]	formalizes the mapping mechanism to Reverse Engineer. It can produce a better quality EER model from given Relational Database schema	very demanding on Input, schema must be in BCNF which is not true in most of systems
Petit et al.[63]	pre-processes Input to Reverse Engineer the system and takes no mandatory assumptions	Lot of Human Intervention is required; user interaction is required
Signore et al.[64]	uses information which is mostly available in a Relational Database, no mandatory assumptions	based on refinement clues, Human Intervention is required
Mfoura[65]	generates Conceptual Schema by analysis of forms	Lot of Human Intervention is required

Few methods are very demanding on inputs which are not available for most of the systems. Still every method has its contribution in one way or the other. In practice one method does not fulfill requirements to reverse engineer a complete system. It is clear from table-2 that a major limitation in most of the methods is involvement of human intelligence; this means the quality of output is directly proportional to skill level of software engineer. Domain knowledge is also required by few methods to carry the process efficiently. Hence an automated process where no or least human intervention is required is still missing.

Table-3 represents the ER constructs wise division extracted by each method. Here "X" represents that the feature is obtained in the output and "-" represents that the method does not retrieve a specific feature of ER or EER model. It is to be noted that three common constructs (entity types, relationship types and attributes) are not listed here. These are extracted by all the methods listed except Blaha's method which generates an OMT model. From this table it is observed that all methods are able to extract primary keys from given inputs. However extraction of primary keys from a system where they are not implemented does not guarantee their correctness, still the confidence on extracted keys is high because they have been extracted by processing of code and data. Human intelligence also plays a vital role in carrying this step. It is to be noted that no method clearly defines way to extract cardinality and participation, also weak entities and multi-valued attributes are not discovered by any of the method except Andersson's method. These are the points which researchers have to work out because if an automated mechanism to get this information is devised then quality of results will improve and reverse engineering process will be more beneficial. Last two columns represent constructs of EER model. Few methods extract generalization and union types from given relational model. Still the process by which these are retrieved needs to be improved because lot of human intelligence is required which affects the quality of results. Table-4 gives another view of previous table (table-3). This table tells about constructs of ER and EER model and from which input the respective construct is extracted.

We can easily tell that which minimal input set is required to generate a construct. It is to be noted that certain constructs are retrieved by data and code analysis which is a tiring job, also human expertise has its role to get certain output. Thus we require an automated model to reverse engineer an existing system and minimize the human intervention.

Strengths and Limitations: Each method has its strengths and limitations. Some require very large number of inputs and some take lot of assumptions which in practice are not true for every system. Also there is least description of analysis of quality of output produced because the major objective of reverse engineering a system is to increase the quality of existing system.

Almost each method requires a human intervention in re-engineering the system which affects consistency of results because the quality of output is dependent on the skill level of the software engineer. In few methods code of the system needs to be analyzed. Availability of code cannot be guaranteed for every system. Also the way the programmer had coded has very high variation. A lot of assumptions are taken in most of the methods which directly affects the quality of the conceptual schema produced. Few methods give steps to map the given relation schema to a conceptual schema but human expertise is required to carry this process. No method defines a complete set of processes which can completely reverse engineer the existing relational schema to a conceptual model in an automated way. Few methods take forms as inputs which are a good source of information but it depends on the designer that how the fields are presented on interface, so the information on forms may mislead the reverse engineer and consequently lead to low quality or wrong results from given system. Hence relying completely on forms for reverse engineering is not an effective technique.

Table-3
EER Constructs Extracted by each method

	Key Attribute	Cardinality	Participation	Weak Entities	Multi valued Attribute	Generalization	Union
Alhajj[2]	X	X	-	-	-	-	-
Andersson[3]	X	X	-	X	X	-	-
Navathe and Awong[4]	X	-	-	-	-	-	X
Chiang et al.[5,6]	X	-	-	-	-	-	X
Johannesson[8]	X	-	-	-	-	-	-
Yeh and Li[9]	X	-	-	-	-	X	-
Premerlani and Blaha[38]	X	-	-	-	-	-	-
Markowitz et al.[62]	X	-	-	-	-	-	X
Petit et al.[63]	X	-	-	-	-	-	X
Signore et al.[64]	X	-	-	-	-	-	-
Mfoura[65]	X	-	-	-	-	-	-

Conclusion

In this paper almost all important DBRE techniques have been analyzed and compared in terms of their capabilities to reverse engineer a given database system. It is observed that input to these methods can be database schema, application code, database instances and user interface forms. Most of methods require more than one type of input. Three methods (Johannesson's, Markowitz et al's, Navathe and Awong's) require only database as input. However, these methods take large set of assumptions on given inputs as listed in Table-1. It has also been observed that every method can reverse engineer basic constructs of an ER model namely entities and relationships, but remaining constructs are not reverse engineered by many methods. Only Anderson's method extracts multi-valued attribute and weak entities. None of the method is able to identify the participation constraints.

Only few methods are able to recover very important constructs like generalization and unions. However, these methods require human interference in the process. It is therefore concluded that a DBRE technique should be developed that can provide support for reverse engineering of the EER constructs – inheritance, participation constraints, weak entities, multi-valued attributes and union type, and carry the process with least human intervention. Such a method can increase the productivity of a reverse engineer resulting into decreased maintenance costs. This is where formal methods can play their role in defining a model with well defined inputs and outputs. Consistency of results will be improved by using a mathematical model; furthermore proposed model will be verified using certain tools. CASE tools must be developed to reduce the time and cost of DBRE. With minimum inputs the tools must have capability to carry the process in an automatic way with least[61].Further work should be done to improve the existing DBRE algorithms in order to reduce the number of inputs. The extent of human intervention required by each method is another factor which should be studied.

Acknowledgements

The authors are thankful to Faculty of Information Technology, University of Central Punjab, Lahore, for providing all the necessary resources to get this work done. The authors are also thankful to Higher Education Commission of Pakistan for encouraging research activities in universities.

Table-4
Input Required for EER constructs

Method By	Key Attribute	Cardinality	Participation	Weak Entities	Multi valued Attribute	Generalization	Union
Alhajj[2]	Relations	Relations	-	-	-	-	-
Andersson[3]	DML	DML, Code	-	DML	DML	-	-
Navathe and Awong[4]	Primary Key	-	-	-	-	Relations	Relations
Chiang et al.[5,6]	Primary Key	-	-	-	-	Data	Data
Johannesson[8]	Primary Key	-	-	-	-	Relations, Functional Dependency, Inclusion Dependency	-
Yeh and Li[9]	Forms, Relations	-	-	-	-	Forms, Data	-
Premerlani and Blaha[38]	Primary Key	-	-	-	-	Relations, Data, Code	-
Markowitz et al.[62]	Primary Key	-	-	-	-	Relations, Inclusion Dependency, Referential Integrity Constraints	Relations, Inclusion Dependency, Referential Integrity Constraints
Petit et al.[63]	Primary Key	-	-	-	-	Relations, Data, Code	Relations, Data, Code
Signore et al.[64]	Relations	-	-	-	-	-	-
Mfoura[65]	Forms, Relations	-	-	-	-	-	-

References

1. Mian N.A. and Zafar N.A. , Key Analysis of Normalization Process using Formal Techniques in DBRE, Proc. of Second Intl. Conf. on Computer Engineering and Applications, *ICCEA 2010*, Bali, Indonesia **(2010)**
2. Alhajj R. and Polat F., Database Reverse Engineering, Proc. Eighth Working Conf. on Reverse Engineering, Stuttgart, Germany, 335-344, **(2001)**
3. Andersson M., Extracting an Entity Relationship schema from a relational database through reverse engineering, *Lecture Notes In CS; 881* ,Proc. of Thirteenth Intl. Conf. on the Entity-Relationship Approach 403-419, **(1994)**
4. Batini C., Ceri S., and Shamkant N., Conceptual Database Design-An Entity-Relationship Approach, *Benjamin/ Cummings*, **(1992)**
5. Chiang R., Barron T. and Storey V., Reverse engineering of relational databases: Extraction of an EER model from a relational database, *Data & Knowledge Engineering*, 107–142, **(1994)**
6. Chiang R., A knowledge-based system for performing reverse engineering of relational databases, *Decision Support Systems* 295–312, **(1995)**
7. Hainaut J. L., Research in Database Engineering at the University of Namur, *ACM SIGMOD Record archive*, **32(4)**, 124 - 128, **(2003)**
8. Johannesson P., A method for transforming relational schemas into conceptual schemas, Proc. of Tenth Intl.. Conf. on Data Engineering, Rusinkiewicz, 115–122, **(1994)**
9. Yeh D. and Li Y., Extracting Entity Relationship Diagram from a Table-based Legacy Database, Proc. of Ninth European Conf. on Software Maintenance and Reengineering, *CSMR 2005*, 72- 79, **(2005)**
10. Jesus L. D., Sousa P., Selection of Reverse Engineering Methods for Relational Databases, Proc. of Third European conf. on Soft. Maint. and Reengineering, 194-197, Amsterdam, Netherlands, **(1999)**
11. Davis K. H. and Aiken P. H., Data reverse engineering: a historical survey, Proc. of Seventh Working Conf. on Reverse Engineering, 70–78, **(2000)**
12. Embley D. W., Liddle S. W. and Lonsdale D. W., KBB: A Knowledge-Bundle Builder for Research Studies, Proc. of Twenty Eighth Intl. Conf. on Conceptual Modeling, Gramado, Brazil, **(2009)**
13. Astrova I. and Stantic B., Reverse Engineering of Relational Databases to Ontologies: An Approach Based on an Analysis of HTML Forms, Proc. of Eighth European Conf. on Principles and Practice of Knowledge Discovery in Databases, 73–78, **(2004)**
14. Mens K. and Tourwe T., Reverse Engineering Aspectual Views using Formal Concept Analysis, CiteSeerX-Scientific Literature Digital Library and Search Engine, **(2008)**
15. Ichise Y. and Ishida Y., Reverse Engineering of Spatial Patterns in Cellular Automata, Proc. of thirteenth Intl. Symp. on Artificial Life and Robotics, Oita, Japan, **(2008)**
16. Cabot J., Gomez C., Planas E. and Rodriguez M. E., Reverse Engineering of OO Constructs in Object-Relational Database Schemas, *JISBD 2008,* **(2008)**
17. Jilani M. A., Aziz A. and Hussain T., An Analysis of Extended Entity Relationship Constructs Extraction in Database Reverse Engineering Approaches, *Science Intl. J.*, **20(4)**, 249-254, **(2008)**
18. Lammari N., Wattiau I. C. and Akoka J., Extracting Generalization Hierarchies from Relational Databases: A Reverse Engineering Approach, *Data and Knowledge Engineering*, **63**, 568-589, **(2007)**
19. Mammar A. and Laleau R., From a B Formal Specification to an Executable Code: Application to the Relational Database Domain, *Intl. J. of Information and Software Technology*, **48(4)**, 253-279, **(2006)**
20. Abbasifard M. R., Rahgozar M., Bayati A. and Pournemati P., Using Automated Database Reverse Engineering for Database Integration, *J. of World Academy of Science Engineering and Technology*, **19**, 1-17, **(2006)**
21. Masoud F. A., Khattab H. T. and Karazoon M. A., University of Jordan Case Tool (Uj-Case-Tool) for Database Reverse Engineering, *J. of World Academy of Science Engineering and Technology*, **9**, 28-31, **(2005)**
22. Ghalayini H. E., Odeh M., McClatchey R. and Solomonides T., Reverse Engineering Ontology to Conceptual Data Models, Proc. of IASTED Intl. Conf. on Databases and Applications, Part of the 23rd Multi-Conf. on Applied Informatics, Innsbruck, Austria, 222-227, **(2005)**
23. Alhajj R., Extracting the Extended Entity Relationship Model for Legacy Relational Database, *Intl. J. on Information Systems*, **28**, 597-618, **(2003)**
24. Datta S. B. and Gupta V.K., Performance Interoperability between RDBs and OODBs, *Res. J.Recent Sci.*, **1 (ISC-2011)**, 419-421, **(2012)**
25. Benslimane S., Malki M. and Bouchiha D., Deriving Conceptual Schema from Domain Ontology: A Web Application Reverse Engineering Approach, *Intl. Arab J. of IT*, **7(2)**, 167-176, **(2010)**
26. Raffat S.K., Siddiqui M.S., Siddiq M. and Shafiq F., Towards the Development of Web-based Ontology Development and Editing (WODE) Tool, *Res. J. Recent Sci.*, **1(12)**, 67-69, **(2012)**

27. Trinkunas J. and Vasilecas O., Building Ontologies from Relational Databases Using Reverse Engineering Methods, Proc. of Intl. Conf. on Computer Systems and Technologies Section II, 6.1-6.6, **(2007)**

28. Gupta D. and Gupta V.K., Approaches for Deadlock Detection and Deadlock Prevention for Distributed systems, *Res. J. Recent Sci*., **1 (ISC-2011)**, 422-425, **(2012)**

29. Jitendra S. and Gupta V.K., Concurrency Issues of Distributed Advance Transaction Process, *Res.J.Recent Sci.*, **1 (ISC-2011)**, 426-429, **(2012)**

30. Neeraj S. and Swati L. S., Overview of Non-redundant Association Rule Mining, *Res.J. Recent Sci.*, **1**(**2**), 108-112, **(2012)**

31. Pannurat N., Kerdprasop N. and Kerdprasop K., Database Reverse Engineering based on Association Rule Mining, *Intl. J. of CS*, **7**(**2**), 10-15, **(2010)**

32. Georgieva, T. Discovering branching and fractional dependencies in databases, *J. of Data & Knowledge Engineering*, **66**, 311–325, **(2008)**

33. Wu A. H., Tan Z. J. and Wang W., Annotation based query answer over inconsistent database, *J. of CS and Technology*, **25 (3)**, 469-481, **(2010)**

34. Soutou C., Relational Database Reverse Engineering: Algorithms to Extract Cardinality Constraints, *Data and Knowledge Engineering*, **28**(**2**), 161-207, **(1998)**

35. Cleve A., Henard J. and Hainaut J. L., Data Reverse Engineering using System Dependency Graphs, proc. of Thirteenth Working Conf. on Reverse Engineering, Benevento, 157 – 166, **(2006)**

36. Blaha M. R., An Industrial Example of Database Reverse Engineering, Proc. of Sixth Working Conf. on Reverse Engineering, 196-202, **(1999)**

37. Premerlani W J., Blaha M. R., An Approach for Reverse Engineering of Relational Databases., Proc. of Working Conf. on Reverse Engineering, **(1993)**

38. Premerlani W J., Blaha M. R., An approach for reverse engineering of relational databases, *Communications of the ACM*, 37, **(1994)**

39. Blaha M. R., Dimensions of Database Reverse Engineering, Proc. of Fourth Working Conf. on Reverse Engineering, 176-183, **(1997)**

40. Moise D. L. and Wong K., An Industrial Experience in Reverse Engineering, Proc. of tenth Working Conf. on Reverse Engineering, **(2003)**

41. Baxter I. D. and Mehlich M., Reverse Engineering is Reverse Forward Engineering, Proc. of Fourth Working Conf. on Reverse Engineering, The Netherlands, 104, **(1997)**

42. Rugaber S., Shikano T., Stirewalt R. E. K. and Carroll L., Reverse Reverse-Engineering, Automated SE Conf., **(2001)**

43. Chu W. C. C., Chang C. H. and Yang S. J. H, Application of the Software Maintenance Predicate/Transition Net (SMPrT-Net) to the Change Control Process of Maintenance, *Intl. J. of Chinese Institute of Engineers*, **23 (5)**, **(2000)**

44. Rugaber S., Stirewalt K., Model-Driven Reverse Engineering, *IEEE Software J.*, **21 (4)**, 45-53, **(2004)**

45. Jahnke J. H., Schafer W. and Ztindorf A., Generic Fuzzy Reasoning Nets as a Basis for Reverse Engineering Relational Database Applications, Proc. of European Conf. on SE, **(1997)**

46. Sneed H.M., An Incremental Approach to System Replacement and Integration, Proc. of Ninth European Conf. on Software Maintenance and Reengineering, **(2005)**

47. Alalfi M.H., Cordy J. R. and Dean T. R., SQL2XMI: Reverse Engineering of UML-ER Diagrams from Relational Database Schemas, Proc. of Fifteenth Working Conf. on Reverse Engineering, 187-191, **(2008)**

48. Chung C.Y., Gertz M. and Sundaresan N., Reverse Engineering for Web Data: From Visual to Semantic Structures, Proc. of Eighteenth Intl. Conf. on Data Engineering, **(2002)**

49. Santoso H. A., Haw S. C. and Mehdi Z. T. A., Ontology extraction from relational database: Concept hierarchy as background knowledge, *J. of Knowledge-Based Systems*, **24**, 457–464, **(2010)**

50. Sousaa P., Jesus L. P., Pereira G. and Abreu F. B., Clustering relations into abstract ER schemas for database reverse engineering, *J. of Science of Computer Programming*, **45**, 137 – 153, **(2002)**

51. Perez J., Ramos I., Anaya V., Cubel J. M., Dominguez F., Boronat A. and Carsi J. A., Data Reverse Engineering of Legacy Databases to Object Oriented Conceptual Schemas, *Electronic Notes in Theoretical CS*, **74(4)**, Elsevier Science, **(2002)**

52. Angyal L., Lengyel L. and Charaf H., An Overview of the State-of-The-Art Reverse Engineering Techniques, Proc. of Seventh Intl. Symp. of Hungarian Researchers on Computational Intelligence, 507-516, Hungary, **(2006)**

53. Henrard J., Hick J. M., Thiran P. and Hainaut J. L., Strategies for Data Reengineering, Proc. of Ninth Working Conf. on Reverse Engineering , 211-220, **(2002)**

54. Muller H. A., Jahnke J. H., Smith D. B., Storey M. A., Tilley S. R. and Wong K., Reverse engineering: A Roadmap, Proc. of Conf. on Future of SE, 47-60, **(2000)**

55. Ali M. R., Why Teach Reverse Engineering ?, *ACM SIGSOFT SE Notes*, **30 (5)**, 1-4, **(2005)**

56. Canfora G. and Penta M.D., New Frontiers of Reverse Engineering, Proc. of Intl. Conf. on Future of SE, **(2007)**

57. Khan, S. A. and Zafar N. A., Promotion of local to global operation in train control system. *J. of Digital Information Management*, **5(4)**, 231, **(2007)**

58. F. M and Khan S. A., Specification and Verification of Safety Properties along a Crossing Region in a Railway Network Control. *Applied Mathematical Modelling J.*, 10.1016/j.apm.2012.10.047, **(2012)**

59. Zafar N. A., and Khan S. A., Towards the safety properties of moving block railway interlocking system. *Int. J. Innov. Comput., Info & Control*, **(2012)**

60. Khan, S. A. and Zafar. N. A., Improving moving block railway system using fuzzy multi-agent specification language. *Int. J. Innov. Computing, Inform. Control*, **7(7)**, **(2011)**

61. Mian N. A., Hussain T., Database Reverse Engineering Tools, Proc. of Seventh WSEAS Intl. Conf. on SE, Parallel and Distributed Systems (*SEPADS '08*) University of Cambridge, Cambridge, UK, **(2008)**

62. Markowitz V. and Makowsk J., Identifying extended entity relationship object structures in relational schemas, *IEEE Transactions on SE*, **(1990)**

63. Petit J. M., Toumani F., Boulicaut J.F., and Kouloumdjian J., Towards the reverse engineering of Denormalized Relational Databases, Proc. of Twelfth Intl. Conf. on Data Engineering, USA, **(1996)**

64. Signore O., Loffredo M., Gregori M., and Cima M., Using procedural patterns in abstracting relational schemata, Proc. of thirteenth Intl. Conf. on Entity-Relationship Approach, *LNCS*, 881 **(1994)**

65. Mfourga N., Extracting Entity-Relationship Schemas from Relational Databases A Form-driven Approach, Proc. of Fourth Working Conf. on Reverse Engineering, 184 **(1997)**