International Journal of Database Theory and Application
Vol.8, No.3 (2015), pp.249-258
http://dx.doi.org/10.14257/ijdta.2015.8.3.21

μE – Automation Framework

Mohammad Almseidin[1], Khaled Alrfou'[2]

[1] *Information Technology, Mutah University, Jordan*
[2] *Information Technology, Tafila Technical University, Jordan*

[1]*mohd@ttu.edu.jo*, [2]*khrfou@ttu.edu.jo*

Abstract

The Entity-Relationship and relational database schema are the most popular methodology for designing database. Several tools have been developed to support drawing an ER - diagram or drawing Relational Database Schema (RDB). This paper aims to develop framework called μE that can help database designers for mapping ER-diagram to Relational database schema and vice versa. μE is an automation framework for mapping. μE framework improves reverse engineering techniques and useful with legacy system. Architecture and implementation of μE framework are present in this paper.

Keywords: *ER-diagram, Relation Database, Handshake XML, μE framework.*

1. Introduction

ER-Draw [1] is an educational prototype tool that used to support drawing ER-diagram in graphical mode. ER-Draw also translates ER-diagram to a relational database schema. Mapping entity relationship to relational databases in graphical fashion is the basic function of the ER - Draw.

A lot of educational universities and companies have legacy database with non-existent documentation. Legacy database with non-documentation is a complex problem for systems need develop continuously. Mapping procedure is always needed for improved database and for best database understanding.

ER-Draw work fine in the normal map (entity relationship to a relational database schema) based on the extract entity type, attributes and relations [1]. Form driven approach [2], support reverse- engineering techniques, for translating relational database schema to entity relationship diagram. Form driven approach based on rules for extract cordiality, relationships after analysis, context meaning [13].

This paper proposed tool called μE to mapping entity relationship toa relational database schema and vice versa. μE is necessary to check your mapping procedure work fine. The main idea of μE is to translate ER-diagram and relational database in single tool based on transforming language. XML is transforming language (database, transform). μE tool use Handshake file, which is an XML file, responsible for the mapping entity relationship to a relational database schema and vice versa.

The remainder of the paper is organized as follows; related work is presented in Section (2) that give a brief discussion about tools used to convert ER-diagram to a relational database schema, and vice versa. Section (3) gives detailed description of μE framework and its components normal and revers transform presented in section (3.1) and (3.2) respectably. μE Pseudo code for normal transform section (3.1.1.) and Pseudo code for reverse transform section (3.2.1.). A real example of how the μE framework can be used on a package is described in section (4) with both normal and reverse transform presented in section (4.1) and section (4.2) respectably, Followed by conclusions and future work in section (5).

ISSN: 2005-4270 IJDTA

2. Literature Review

The ER model was introduced as a tool for data modeling by P. Chen in 1976 [1]. Translate from ER diagram to relational database and vice versa discussed by Shuyun Xu, Yu Li, Shiyong Lu develops prototype called ER-draw [4] that work for a mapping entity relationship diagram to a relational database schema.

ER-draw based on XML diagram and this tool support object oriented techniques [1]. Representing ER-diagram as XML discussed by P. Chen [2] on the other hands, N. Mfourga proposed framework as part of relational database reverse engineering. Framework translates ER from a relational database schema [5].

Downing Yeh a, Yuwen Li are present approach that is called DBRE support extract extended entity-relationship diagram from a legacy database [10]. On the other hand Valerio Cosentino and Salvador Mart´ınez proposed approach for mapping UML/OCL [9] from relational database Yeh, D. focus on study how to use reverse engineering with legacy database [14]. W. Premerlani focuses on using UML in reverse engineering techniques [3].

ER diagram enhancements are main topics for designers and how make an ER diagram easy to build. Designers need enhancement for ER diagram. Semi–automatic assessment of conceptual database diagrams is the main topic for BATMAZ, F. and HINDE present drawing tool compare student solution with optimal solutions and help examiner to mark [7]. Peter Pin-Shan Chen proposed method for mapping from English sentence to ER diagram [11]. Using natural language based one structure and contextual meaning of the English language, such as a common noun, transitive verb, *etc*. [8]. Helen C. Purchase, etl concerned about which notation type (chain model and logical data structure) is very useful and easy to use [8].

P. S. Dhabe, etl study articulated entity relationship (AER) to complete automation normalization. [6]. David W. Embley, *et al.* proposed tool called SQL2ORM (Object Relationship Model). SQL2ORM [12] approach precedes automation correct transformation to high-level view. SQL2ORM produce quicklytranslates from initial model. Then systematically develop a final model by rearrangement entity, relationship and map the initial model to conceptual view.

Yang Dongqing and its researcher proposed Ana practical solution for the modeling of real-world problems using first the extended ER model and then its refinement into the relational model [15]. Yen-Ting Chen and *etc*. They propose an algorithm to design of methodologies for building multidimensional models based on source ER diagrams, and it contains incorporating two features, namely, grain preservation and the minimal distance from each dimension table to the fact table [16].

The reverse engineering technique proposed by Alalfi, M.H and *etc*. researchers developed an approach and a tool to help bridge the gap between application and data modeling based on source transformation technology. Both data and application models can be manipulated using the same UML-based tools [17].

3. μE Framework Overview

The main structure of the proposed μE framework is shown in (Figure 1). Structure of μE Framework based on recognizing XML file that generated from μE-GUI. Recognize XML file which reading first tag to specify the type of map. XML file are used as input for analytical process in two types of mapping. A new XML file will be generated from the previous analyzer process as output. Visualizer process receives XML file that generated from the analyzer process as input. Visualizer process draws ER-diagram or relational database schema based on XML constraints tags.

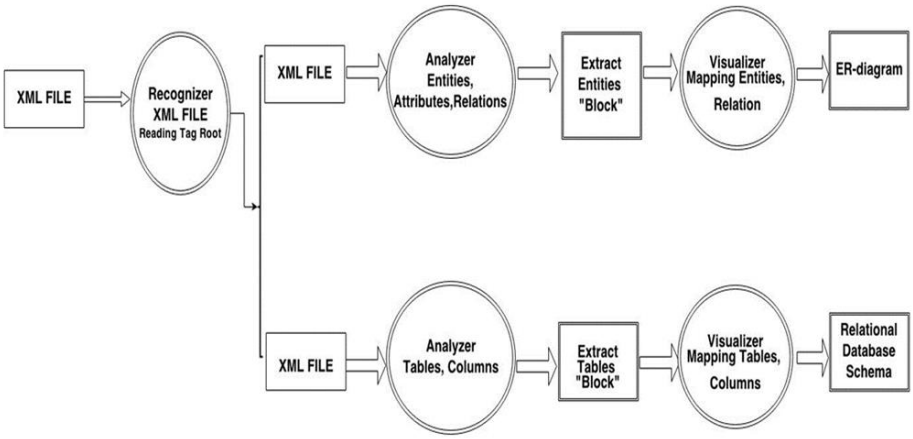

Figure 1. μE Framework Processes

3.1. Normal Mapping (ER-diagram to Relational Database)

Normal transformation is transforming from ER diagram to a relational database schema using μE framework. Recognize process use XML file as input to specify type of XML (ER-diagram, relational database). Analyze process state after XML file received with root tag equal "ER "analyze process, extract entity type, attributes, relations and cardinality.

The main function of the mapping process applies the following rules if no constraints between entity set, assign entity in a separate table. If an entity set has key constraints with relationship, limited total participation, combine tables, that contains entity set and relationship. If an entity has key constraints with relationship, plus total participation, create, combine table that contains entity with a relationship with NOT NULL. If the entity set has a total participation with no key constraints, μE enforce constraints. Output of the mapping process generates relational database schema.

3.1.1. Normal Mapping (ER-diagram to Relational Database)

The following pseudo code represents a normal transform steps based on μE framework. The database designer draws an ER - diagram using μE-GUI to create XML files. Recognize function receives an XML file as input to specify type of map. Analyze function used XML file as input and extract entities, attributes in one block for each entity.

Mapping function received XML block and extract relation and cardinality between two entities. Depend on type of cardinality and relationship type, XML file will generated with constraints to create primary key and foreign key. Draw function used the XML block as input and draw relational database schema tables based on XML tags, constraints, primary keys and foreign key.

```
1   Input (X);
    // load XML file
2   Recognize (x);
    // recognizing function used to specify type of transforms.
3   If recognize (x) == 1 then
    //For normal transform always recognize(x)==1
4   Analyze(x);
    // analyze function used to extract entities and attributes, and adding every entity
    // with its attributes in one block
```

```
5    Mapping(x);
     // mapping function used to find relationship and cardinality between two entities
6    If (cardinality ==1) then
     // if the value of cardinality =1 then return 1 and the cardinality is 1-1
7    Add-PK (source, target);
     // add primary key from source as foreign key to target.
8    Else if (cardinality ==2) then
     // if the value of cardinality=2 then return 2 and the cardinality is 1-m.
9    Add-PK (source, target);
     // add primary key from source as foreign key to target.
10   Else if (cardinality ==3) then
     // if the value of cardinality=3 then return 3 and the cardinality is n-m.
11   Create (source , target );
     // create function used to create new table with fields primary keys from source and destination
12   End if
13   Draw relational database schema;
14   End if
```

3.2. Reverse Mapping (Relational Database to ER- Diagram)

The reverse transformation is transforming from a relational database schema to ER diagram using μE framework. Recognize process used XML file as input to specify type of XML (ER-diagram, relational database). Analyze process start after receiving an XML file with root tag equal "DBR ".Analyze process deal with reverse transformation based on extract tables, columns after defining each table and its columns. Mapping process starts after receiving an XML file from the analyzer process as input. Mapping process applied following rules cardinality extraction, relationship extraction, normalization and schema integration to create XML file contain constraints to draw an ER - diagram.

3.2.1. Pseudo Code for Reverse Transform

The following pseudo code represents reverse transform steps based on μE framework. The database designer draw relational database schema tables using μE-GUI to create XML files. Recognize function receives an XML file as input to specify type of map. Analyze function used XML file as input and extract tables, columns in one block for each table.

Mapping function received XML block and extract tables and cardinality between two tables. Depend on type of cardinality and relationship type. Draw function used the XML block as input and draw relational database schema tables based on XML tags, constraints, primary keys and foreign key.

```
1    Input (X);
     // load XML file
2    Recognize (x);
     // recognizing function used to specify type of transforms.
     if (recognize (x) ==2) then
     //For reverse transform always recognize(x)==2
3    Analyze(x);
     // analyze function used to extract tables and columns, and adding each table
     // with its columns in one block
4    Mapping(x);
     // mapping function used to find relationship and foreign keys between two tables
5    If (re-cardinality==1 ) then
     // the value of re-cardinality ==1 then return 1 and the cardinality is 1-1
```


6 Remove-fk(table1, foreign key);
//Remove function used to remove foreign key from table1
7 Find-relation (table 1, table2);
// find-relation function used to specify cordiality of relation based on foreign key location on tables
8 Else if (re-cardinality==2)
// the value of re-cardinality ==2 then return 2 and the cardinality is 1-m
9 Remove-fk (table2, foreign key);
//Remove-fk function used to remove foreign key from table2 that has many cardinality
10 Else if (re-cardinality==3)
// the value of re-cardinality ==3 then return 3 and the cardinality is m-n
11 Remove-table (table);
//Remove table function used to remove foreign key from table2 that has many cardinality
13 End if
14 Draw ER-diagram;

4. Employee Database Example

Suppose that employee in the company has following attributes employee name, employee identifier, and employee salary, all of these attributes organized as follow (name, emp_id, salary).company also include department with the following attributes department name, department identifier and all of these attributes organized as follow (dept_name, dept_id).

Suppose that cardinality is 1:M, Database designer need to transform employee example in both normal transform and reverse transform using µE framework.

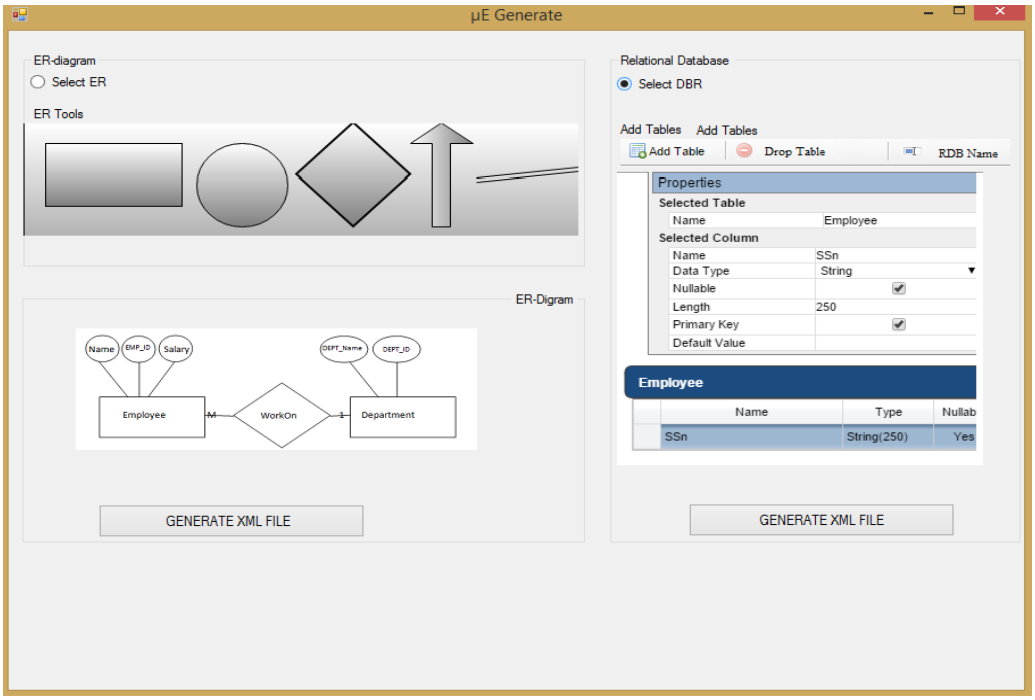

Figure 2. Generate XML File for ER-diagram

4.1. Normal Transformation Example

µE-graphical user interface generator shown in (figure 2). Which include two main operations to help database designers create XML for ER-diagram and relational database

schema tables. Normal transformation is transforming from ER diagram to a relational database schema using μE framework. For employee example database designer need to select ER-diagram and then draw ER-diagram using μE-GUI. After that click to generate an XML file that will be used as input for μE framework.

μE-graphical user interface generator used to draw and generate ER-diagram XML file that shown in (figure 3). ER-diagram XML file start with first entity name employee as the first tag with attribute values organized as follows (emp_name, emp_id, salary), primary key value equal (emp_id).

μE GUI extracts second entity name, department that shown in (figure 3) from employee example, as second tags with attribute values organized as follows (dept_id, dept_name), primary key value equal (dept_id).

The relation tag that shown in (figure 3), it contains the value of relation name (work_on) and total value of the employee entity with a partial value of department entity. Cardinality tag has two values for cardinality name and type as follows (one: many).

```
<?xml version="1.0"?>
- <ER name="company">
  - <Entity name="employee">
      <attribute name="emp_id" value="primaryKey"/>
      <attribute name="emp_name"/>
      <attribute name="salary"/>
    </Entity>
  - <Entity name="department">
      <attribute name="dept_id" value="primaryKey"/>
      <attribute name="dept_name"/>
    </Entity>
  - <relation>
      <member name="employee" value="total" rel="work_on"/>
      <member name="department" value="partial" rel="work_no"/>
    - <cardinatlity>
        <card name="employee" value="one"/>
        <card name="department" value="many"/>
      </cardinatlity>
    </relation>
</ER>
```

Figure 3. ER-diagram XML File

4.2. Reverse Transformation Example

μE-graphical user interface generator shown in (figure 4), which used to help database designers create XML for relational database schema. The reverse transformation is transforming from a relational database schema to ER diagram using μE framework. For employee example database designer create the first table for the employee with the following columns (emp_name, emp_id, salary). Database designer creates a second table for the department with the following columns (dept_id, dept_name). Database designers must specify a primary key value and foreign key value using μE GUI. After that click to generate an XML file that will be used as input for μE framework.

μE-GUI generator used to draw and generate relational database schema XML file that shown in (figure 5). relational database schema XML file start with first table name employee as the first tag with column values that organized as follows (emp_name, emp_id, salary), primary key value equal (emp_id).

μE GUI extracts second table name, department that shown in (figure 5) from employee example, as second tags with column values organized as follows (dept_id, dept_name), primary key value equal (dept_id) and foreign key value equal (emp_id).

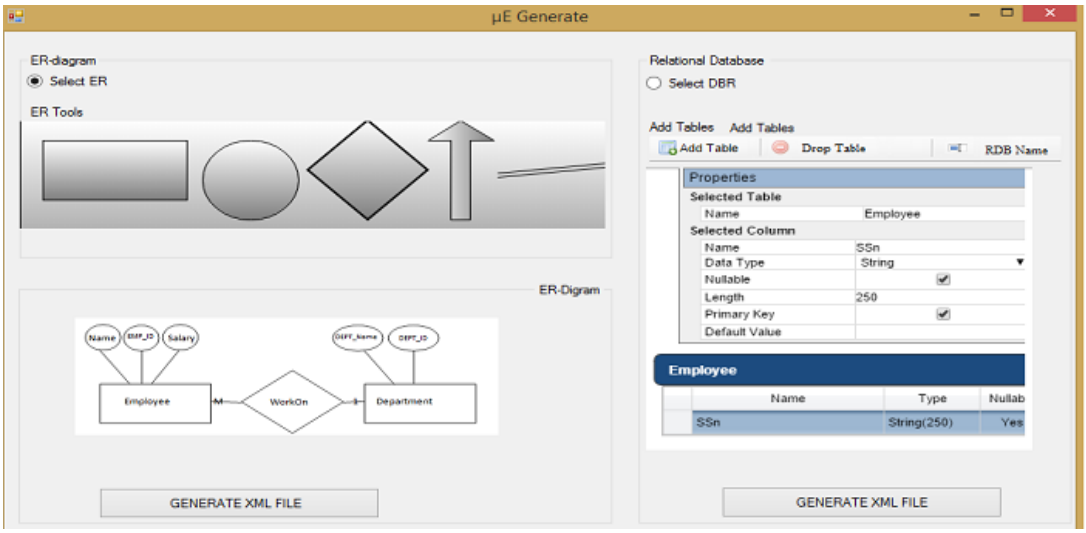

Figure 4. Generate XML file for RDB

```
<?xml version="1.0"?>
- <RDB name="company">
  - <table name="employee">
      <col name="emp_id" value="primaryKey"/>
      <col name="emp_name"/>
      <col name="salary"/>
    </table>
  - <table name="department">
      <col name="dept_id" value="primaryKey"/>
      <col name="dept_name"/>
      <col name="emp_id" value="foreignKey" ref="employee"/>
    </table>
</RDB>
```

Figure 5. RDB XML File

5. Conclusion and Future Work

Mapping form ER diagram to relational database and vice versa, is very important for database designers. Some of legacy system with none documentation needs a useful tool to extract relational database schema. Working with legacy system will save costs.

µE framework proposed in this paper deal with time saving, reduce complexity, and improve reverse engineering techniques. Database designers have more chance to check the validity and completeness of mapping procedure. µE framework based on transformation database language (XML) and handshake maker tool. Cardinality and relations are difficult to measure in relational database schema mapping process. Complex database also is difficult to mapping using the µE framework because contains a lot or constraints.

As part of the future work µE framework will be applicable to read and detect any types of file not only XML file. Increase artificially by reading selected image, to identify and recognize the type of input (ER or relational database diagram). Generate initial SQL statements with DBR that give database designers overview of tables that should be exist.

Acknowledgements

First and foremost all praises and thanks to Allah for giving us the ability to write this paper. We would like to thank Dr. Mustafa Hammad, for the guidance and support. Without his helpful suggestions, advice and encouragement, this work would not have been possible. We would also like to thank our friends and colleagues for all discussions and advices that we have shared throughout our paper. We are very grateful to our family and our warmest thanks to our dear mother for their support, patience, and love. Finally, we would like to thank all who has direct and indirect support and helped.

References

[1] P. Chen, the Entity-Relationship Model - Towards a Unified View of Data, ACM Transactions on Database Systems, 1(1):9-36, **(1976)**.
[2] P. Chen, ER Model, XML and the Web, International Conference on Conceptual Modeling (ER), pp. 538, 1999.
[3] W. Premerlani and M. Blaha, An approach for reverse engineering of relational databases, Comm. of the Assoc. for Computing Machinery, 37(5):42{49} May (**1994**).
[4] Shuyun Xu , Yu Li, Shiyong Lu , ERDraw: An XML-based ER-diagram Drawing and Translation Tool, International Journal of Database Management Systems. **(2010)**.
[5] N. Mfourga, "Extracting Entity-Relationship Schemas from Relational Databases" , A FormDriven Approach, Proceedings of the 4th Working Conference on Reverse Engineering **(1997)**.
[6] P. S. Dhabe, Dr. M. S. Patwardhan, A. A. Deshpande, M. L. Dhore, B.V. Barbadekar and H. K. Abhyankar, "Articulated Entity Relationship (AER) diagram For Complete Automation Of Relational Database Normalization, International Journal of Database Management System (IJDMS), vol.2, no.2, **(2010)**.
[7] F. Batmaz and C. J. Hinde, A Diagram Drawing Tool for Semi–Automatic Assessment of Conceptual Database Diagrams, 10th CAA International Computer Assisted Assessment Conference on 4th and 5th July, **(2006)**; Lougborough University.
[8] H. C. Purchase, R. Welland, M. McGill and L. Colpoys, Comprehension of Diagram Syntax: An Empirical Study of Entity Relationship Notations, International Journal of Human-Computer Studies, vol.61, no.2, **(2004)**, pp187-203.
[9] V. Cosentino, and S. Mart´ınez, "Extracting UML/OCL Integrity Constraints and Derived Types from Relational Databases", AtlanMod team, Mines Nantes & INRIA & LINA, **(2012)**.
[10] D. Yeh, Y. Li and W. Chu, "Extracting Entity-Relationship Diagram From a Table-Based Legacy Database", The Journal of Systems and Software 81 (764–7716) **(2008)**.
[11] Peter Pin-Shan chen, English Sentence Structure And Entity Relationship Diagrams, University of California, Los Angeles, **(2004)**.
[12] D. W. Embley and M. Xu, "Relational Database Reverse Engineering: a ModelCentric, Transformational, Interactive Approach Formalized in Model Theory, Database and Expert Systems Applications", Proceedings, Eighth International Workshop, **(1997)**.
[13] M. Anderson, "Extracting an entity relationship schema from a relational database through reverse engineering", In: ER International Conference, **(1994)**.
[14] D. Yeh, Y. Li, W. C. Chu, "Extracting entity-relationship diagram from a table-based legacy database", Journal of Systems and Software, vol.81, no.5, **(2008)**, pp.764–771.
[15] Y. Dongqing, T. J. Teorey and J. P. Fry, "A practical approach to transforming extended ER diagrams into the relational model", Information Sciences, vol.42, no.2, **(1987)**, pp.167–186.
[16] Y.-T. Chen and P.-Y. Hsu, A grain preservation translation algorithm: From ER diagram to multidimensional model, Science direct, **(2007)**.
[17] M. H. Alalfi and J. R. Cordy, SQL2XMI: Reverse Engineering of UML-ER Diagrams from Relational Database Schemas, Reverse Engineering, WCRE '08, 15th Working Conference, **(2008)**.

Authors

Khaled Alrfou', he is a researcher at Tafila Technical University. In 2014 was graduated master computer science/Artificial intelligence at Warsaw University of Technology. His areas of interest are Knowledge Representation and Reasoning, Software Engineering, Neural Networks, Machine Learning, Big data and database management system.

Mohammad Almseidin, he is a student and a researcher for thesis at the Master's degree in Computer Science at Mutah University. His Bachelor's stage ended in 2005 with distinction majoring in computer science at Al-Hussein Bin Talal University. In 2013 began in the master's degree at the Mutah University. His areas of interest are Software Engineering, Computer Networking, Big data and database management system.

