

#EER WEB – ONLINE DATABASE DESIGN TOOL FOR INTRODUCTORY DATABASE COURSES

MILOŠ BOGDANOVIĆ

Faculty of Electronic Engineering, University of Niš, Serbia, milos.bogdanovic@elfak.ni.ac.rs

LEONID STOIMENOV

Faculty of Electronic Engineering, University of Niš, Serbia, leonid.stoimenov@elfak.ni.ac.rs

Abstract: Most universities which incorporate informational technologies and computer science studies have an introductory course dealing with the development and design of databases. In order to facilitate the database design learning process, students are often provided with a database design tools. A majority of publicly available database design tools represent general purpose tools which do not support conceptual database modelling process or do not follow the notation used in the course. In this paper, we will present a tool developed with educational purpose – to make the process of relational databases design easier for the students and partially automate it. This tool is named #EER Web and it has evolved due to the experience in using its predecessor called #EER - a desktop database design tool that we have previously developed for educational purposes. #EER Web enables fast and efficient development of the relational database conceptual model and its automated compilation into a relational model and further to data definition language (DDL) commands. #EER Web tool is based on the extended entity-relationship (EER) model for conceptual modelling of relational databases. It is an AJAX-based Web application whose modular architecture along with the benefits that its usage brings, will also be presented.

Keywords: database design, introductory courses, conceptual modelling, Web

1. INTRODUCTION

Nowadays, there is hardly any software system that doesn't store particular information in some kind of a database. In contemporary information systems, databases are considered to be ubiquitous. Therefore, databases very often have a crucial role in the development of software systems. Since a single database can be shared between different applications, the process of database designing is extremely important as it needs to provide the solutions that could respond to different and often opposite requirements.

Due to the high importance which databases have in contemporary software systems, introductory database courses are sometimes considered core topic in Computer Science curriculum. Therefore, significant attention should be devoted to theoretical concepts of database development process. In order to focus attention of students to the phase of conceptual database design, they should be provided with an automated database design tool. Concerning its functionalities and capabilities, this tool would facilitate the overcoming of the theoretical concepts of database design to the students. The main aim of introductory database courses is comprehension of the database design process – the process of generating a model of a database using a specific data model. The process of database design itself is divided into four phases [1][2]:

- Collect and analyze demands,
- Conceptual design,
- Logical design,

- Physical design.

For the purpose of understanding the process of generating a model of a database, the phase of conceptual design is probably of the highest importance. In this phase the conceptual database model is designed to be independent from any kind of implemental details (hardware and software platforms, chosen DBMS, programming languages, client applications). The output of this phase is a conceptual database model which consists of the detailed description of entities, relationships, and constraints.

While developing conceptual database design, different conceptual database models can be put into use. The basic purpose of the conceptual model is to provide means for the description and formulation of the database design demands. These demands have been collected in the previous phase. The importance of a conceptual database model can be observed through its capability to prepare the foundation for the following phases in the process of database designing.

There appears to be more support for Entity-Relationship (ER) modelling than for UML modelling in the introductory database course. Most of the Computer Science (CS) and Information Systems (IS) university curricula's use traditional ER modelling techniques as their primarily modelling technique in their undergraduate introductory database courses [3]. As stated in [3], the complexity of UML class diagrams along with the primary focus on relational databases in introductory database courses are compelling reasons to consider using

Extended Entity-Relationship (EER) notation in the class instead of UML.

The rest of the paper is organized as follows: in the section *Database Design Tools – introductory database courses alternatives*, the characteristics of similar tools (desktop and Web solutions), the target groups of the users and the arguments for the construction and usage of this kind of tool are presented. In the section *#EER Web – Online Relational Database Design Tool*, the characteristics of our tool are presented. The section *The Architecture of #EER Web* describes the general architecture of the tool. The section *The Usage of #EER Web for Educational Purposes* describes the way the tool will be used in Faculty of Electronic Engineering Niš.

2. DATABASE DESIGN TOOLS – INTRODUCTORY DATABASE COURSES ALTERNATIVES

Conceptual database model has a significant role within introductory database courses because of its great importance for the database design process. Any software tool which aids the process of conceptual database model development, if used within education courses, could be significant for student's education. The usage of such tool can be very helpful if student are encountering database development for the first time. This situation often happens while attending introductory database courses. The idea behind taking advantage of this kind of a tool is to focus the attention of students to conceptual database design. This will in turn cause students to spend more time on understanding the concepts of database design. Also, it would be advisable for universities to adjust the (E)ER graphical notation use within tool with the ones uses in lectures.

Considering the development of a tool that would be specially developed for educational purposes and therefore adapted for certain courses, the question that appears is whether some of the existing tools can be uses with the same educational effectiveness? In order to emphasize the advantages of the custom developed tools, we will present a short list of some of the most frequently used tools and their characteristics. Currently, there is a significant number of both commercial and freeware tools which support database design process. Characteristics of these tools vary from complete support for the database design process to a support limited to particular database design phases.

One of the best and most frequently used commercial tools is Microsoft Visio (<http://office.microsoft.com/en-us/visio/>). Visio represents a tool which belongs to a group of tools designed for professionals involved in the process of development and usage the different types of database management systems (DBMS). As such, Visio can be considered quite complex. Therefore, its usage for educational purposes can be questionable since a majority of users (students) is encountering the database design process for the first time.

The following shareware and freeware tools are probably better choice for educational purposes. It is our opinion that these tools are less complex and easier to use:

- **Toad Data Modeler** – a freeware tool which enables the development of ER diagrams, considers individual database options such as referential integrity, constraints, domains, triggers, stored procedures, functions and views. It also enables reverse engineer of already existing database structures, generation of complex SQL scripts automatically along with HTML or RTF reports (<http://www.casestudio.com/enu/default.aspx>) [4],
- **WWW SQL Designer** – online Web-based database schema design tool. It enables: drawing of ER designs, editing of tables and rows, management of keys, creation of relations (FK constraints), design storing and loading and import of DB schemas (<http://ondras.zarovi.cz/sql/demo/>) [5],
- **DB Schema Editor** – a free Web application which enables designing of database schemas and generation of scripts for SQL Server, My SQL, Postgre SQL, SQLite, Firebird and Oracle directly from browser. It is free, easy to use, multiengine and online Web application (<http://www.dbschemaeditor.com/OnlineDB.aspx>) [6].

Intelligent tutoring systems (ITS) are envisioned as the next generation of database design tools. The goal of these systems is to provide individualized pedagogical sessions to all students [7]. KERMIT – the Knowledge-based Entity Relationship Modelling Intelligent Tutor is one of the well known solutions from this category [8]. This system is developed as a problem-solving environment. KERMIT solution allows students to practice their ER modelling skills with the individualized assistance of the system. The solution presented in this paper also assists student in the database design process and can be classified into this group of tools. However, our tool implements only the basic assistance development. It does not allow students to make a mistake while developing conceptual database model it terms of misconnected (E)ER diagram elements.

A majority of the described tools is designed to cover a large number of different database design approaches. Also, these tools tend to cover a large number of existing DBMSs. These tools are designed for database experts and professionals and can be very complex and immense. These characteristics suggest that the appliance of these tools in educational process can be very difficult and requires additional activities which are related to the special training for usage of a particular tool.

Another common characteristic of these tools is that they support (E)ER model and its graphical notation as suggested in [9] or UML modelling techniques (<http://www.uml.org/>) [10]. If an alternative notation should be used, users would have to use or develop plug-ins for the selected tool which will then allow the usage of the desired notation.

3. #EER WEB – ONLINE RELATIONAL DATABASE DESIGN TOOL

The motivation for the development of the #EER Web tool, whose architecture is shown in this paper, was to help overcome the problems faced by users who are encountering database development for the first time. Also, our aim was to develop a Web-based solution that can be used for the development of a relational database conceptual model using the EER modelling technique. By providing users with on-line available EER model development application, the process of conceptual database model development can be simplified and faster. Since this tool was developed for the educational purposes, the target group of users are student which study Computer Science. A majority of these students encounters the problem of relational database model developing for the first time while attending introductory database courses. We consider Computer Science students to be a group of users who have advanced knowledge about usage of computers and have experience in drawing various types of diagrams. Therefore, we envisioned that this tool will provide students with means to develop relational database model without unnecessary and complex functionalities.

#EER Web is the tool which tends to partially automate the process of conceptual database model development

and usage. It also tends to make this process easier to understand for the student by focusing their attention towards the conceptual database model development rather than the development of the relational model and DDL commands. Since ER model is widely used within introductory database courses, we were encouraged to choose Extended Entity-Relationship model and its graphical notation in the form of EER diagram. The graphical notation implemented in the #EER Web follows the notation used in lectures and is suggested by Elmasri and Navathe [2] (See Image 1.)

The tool is implemented in the form of mini developing environment used for the relational database design. According to its purpose, this tool supports the database development from the phase of the creation of database conceptual model until the phase of automatic generation of commands used for creation of data (DDL commands) at the end of the designing phase.

#EER Web is developed as an AJAX-based Web application. The tool provides a wide range of functionalities giving the support to the process of relational database designing.

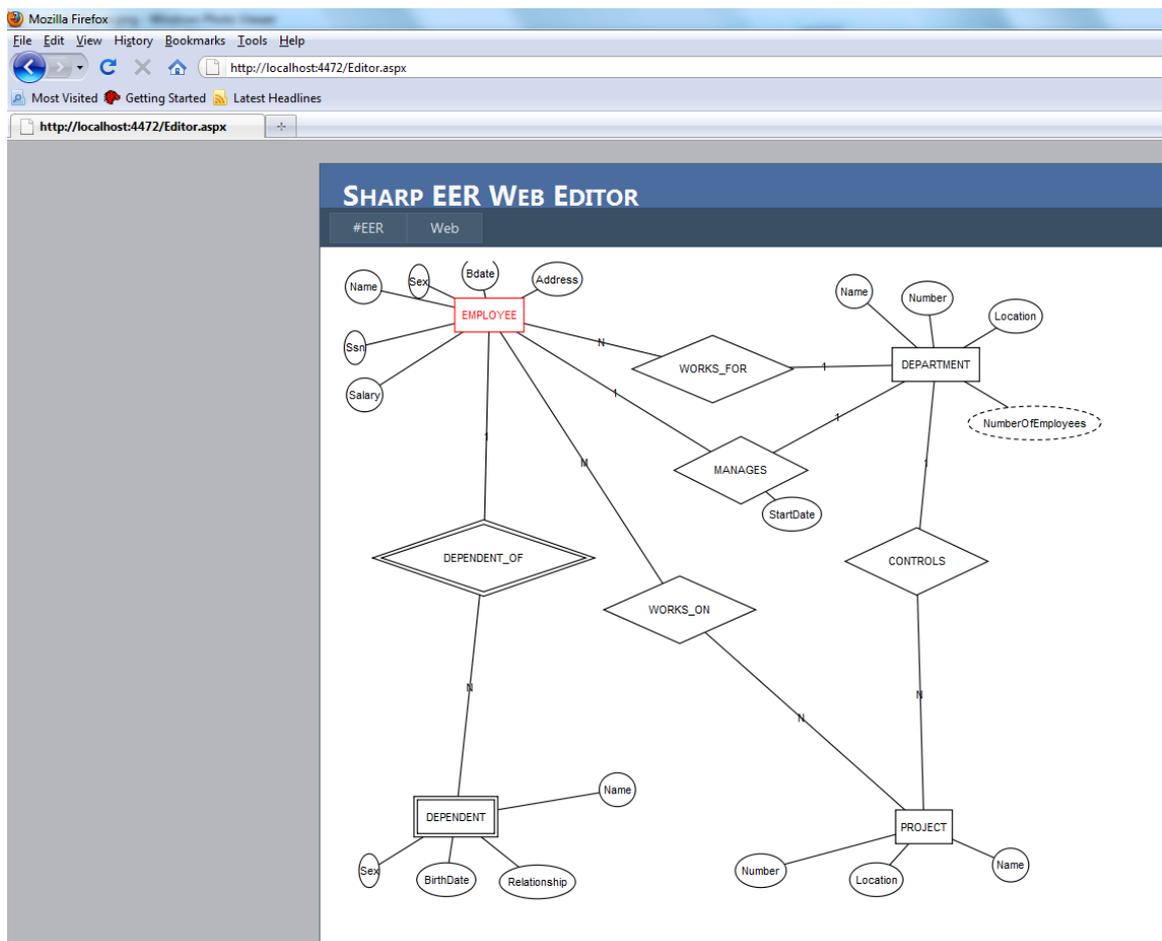


Image 1: A part of EER diagram of the COMPANY database (example taken from Elmasri & Navathe, 2002)

These functionalities can be divided into the following groups:

- *Drawing EER diagram* – fast and easy outlining of the complete EER database diagram, the selection of the individual elements or group of elements in EER diagram, the simple manipulation and modification of the diagram elements etc.,
- *Automatic translation of EER model into relational model* – automated compilation of EER model in relational model and generation of relational diagram underlying derived relational model,
- *Creation of DDL commands* – automatic generation SQL commands based on derived relational model.
- *Export of the EER diagrams into standardized graphical formats* – the possibility to export EER diagrams in TIFF, JPG, BMP, PNG, EMF, WMF
- *Export of the Generated DDL commands into TXT file* – generated SQL commands can be saved in the textual file for the possible processing and usage in database development

4. THE ARCHITECTURE OF #EER WEB

The general architecture of the tool, shown in Image 2, could be divided into client and server part of the application. The client application is #EER Web Editor which allows drawing of database EER model. #EER Web Editor was created using object-oriented JavaScript programming language concepts.

IO module accepts user actions (EER diagram creation) on the client side of the applications. This module controls a drawing surface used for the creation of a database conceptual model.

#EER Web creates an internal object model of the drawn EER database model. The internal object model contains a single object for each of the elements of the drawn EER database model. Each object contains properties which describe the characteristics of the EER model element: element type, position, connections, parts, etc. Object builder module is responsible for the creation of objects which represent parts of the EER database model. In order to assist users when creating the EER database model, Object builder module implements build logic which ensures the proper connectivity of EER diagrams

elements. Build logic consists of a set of rules which define the type of the connections which can be used for the connectivity of particular elements. In this way, #EER Web tool helps users by disallowing incorrect element connections and provides properly drawn EER database diagram.

The visualization of the EER diagram is performed by the Drawing module on client side of the application. Drawing module uses Raphael JavaScript library [11] which simplifies the work with vector graphics on the Web. This library uses the SVG W3C Recommendation [12] and VML [13] as a base for creating graphics. Each of the objects within internal object model of the EER database diagram contains a method used for the drawing of its graphical representation. Object builder module provides objects to the Drawing module which invokes *Draw()* method of each of the objects contained within the internal object model of EER diagram.

After the EER database model has been created, users have the ability to translate the EER model into the relational model. Also, on the basis of the resulting relational model, #EER Web tool generates DDL statements that can be used to implement the database in various DBMSs. The functionality of translating EER model into relational model and functionality of generating DDL statements on the basis of relational model are delegated to the server side of #EER Web tool. Internal object model of EER diagram is transformed into XML document on the client side of the application. Created Xml document contains description of the generated model. This XML description is sent to the server side, where all requests are accepted by Request dispatcher Web service. Request dispatcher Web service distributes accepted documents to individual server-side modules depending on the type of the request. In the case of requests for translation of EER model, dispatcher Web service forwards the received description to the Object Model Builder module which performs the reconstruction of EER diagram object model. Reconstructed model is forwarded to the Translator module which performs transformation of an object model into relational database model.

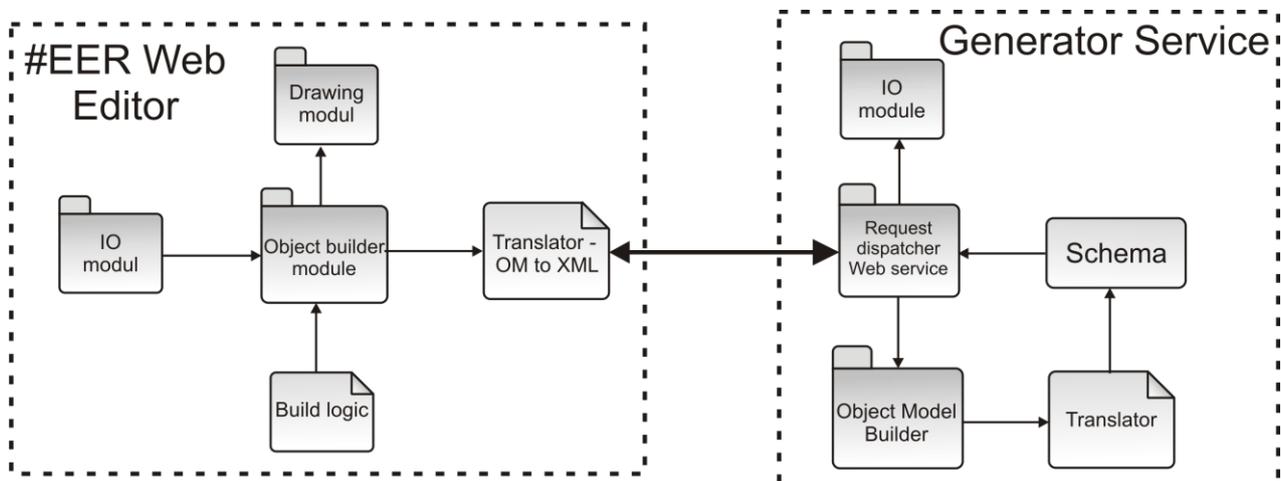


Image 2: The architecture of #EER Web database design tool

Afterwards, the generated relational model is transformed into an XML document which is returned to the client side of the #EER Web tool which will perform its visualization.

In addition to functionalities of relational database model and DDL commands generation, the #EER Web tool users have the ability to permanently store created EER models, as well as export created models into standardized graphical formats. These functionalities are implemented in the IO module of the application server side.

5. THE USAGE OF #EER WEB FOR EDUCATIONAL PURPOSES

The presented tool is the successor of the similar desktop application which we have previously developed [14][15]. The previously developed desktop application has been used in the past five years as an auxiliary educational resource in the Faculty of Electronic Engineering, University of Nis. Our aim is to substitute this tool with #EER Web within the introductory Databases course. This course usually gathers 100 students and consists of obligatory laboratory exercises, projects, and final exam. #EER Web will be used by both teachers and students. Teachers will use in order to present basic concepts of EER database model. We expect this tool to provide student and teachers with all advantages offered by the previous desktop version, e.g. to provide them with ability to:

- learn basic concepts of EER database model and develop EER diagrams,
- translate EER database model into relational model,
- translate relational model into DDL commands.

Since #EER Web incorporates a mechanism to translate the EER model to relational model and a relational model to DDL commands, this tool will provide student with opportunity to learn translation methods from any place which has Internet access. It does not require any kind of installation and can be used from any publicly available Web browser.

The usage of the previous version of this tool (desktop application) was obligatory in laboratory exercises. On the other hand, students had the opportunity to choose the tool they would use in order to develop their projects. Our statistics shows that over 50% of students used the previous version of our tools in their projects. Since both versions use the format of the developed EER model when storing it, created models can be easily exchanged between these tools. Further, by making this tool on-line available, we expect the usage percentage to increase in the following years.

6. CONCLUSION

Contemporary commercial or freeware tools are designed to be adaptable to various project requirements. Since they are designed for IT professionals, their usage for educational purposes can bring as many advantages as perplexities. On the other hand, tools which are

specialized for a particular field of usage can bring a number of advantages. Usually, these tools are more optimized, intuitive at the level of the problem it is dealing with, easy to use for the designers familiar with the notation the tool uses. We consider #EER Web to be such tool. This tool addresses the problem of the development of a specific type of databases – the relational databases. It is used by students and tends to avert all the attention towards the process of conceptual database model development. Therefore, it directs all its users towards the narrow extent of the possible technologies and notations. Being a Web-based solution, we consider it to be easy to overcome and use, especially for students who are familiar with the notation #EER Web uses. This kind of tool gives students the chance to test their knowledge of the database design in an easy and fast way.

Since the #EER Web has been developed for educational purposes primarily, its usability would be significantly increased if it could be integrated with learning management systems (LMS) which exist within an educational institution. #EER Web integrated with LMS would give students an opportunity to use LMS modules, such as forums, chat, scheduler, etc., while working on their database design project. In this manner, students could share their knowledge and experience with their colleagues and organize their activities through a single access point – LMS that they are already familiar with. We expect that this kind of future integration will lead towards a synergetic learning phenomenon and speed up the learning process.

ACKNOWLEDGEMENT

Research presented in this paper was funded by the Ministry of Science of the Republic of Serbia, within the project "Technology Enhanced Learning", No. III 47003.

LITERATURE

- [1] Connolly, T., Begg, C., *Database systems (4th ed.)* ISBN 0-321-21025-5, Addison-Wesley, 2002
- [2] Elmasri, E., Navathe, S., *Fundamental of database systems (4th ed.)* ISBN 0-321-1222-6. Addison-Wesley, 2002
- [3] Suleiman, J., Garfield, M., *Conceptual data modelling in the introductory database course: Is it time for UML?*, Journal of Information Systems Education, 17(1), 93-100, 2006
- [4] Quest Software, *Toad Data Modeler*, 2007 http://www.casestudio.com/enu/database_design_free_ware.aspx
- [5] Zara O., *WWW SQL Designer*, 2010 <http://code.google.com/p/wwwsqldesigner/>
- [6] Botella B. D., Diez M. J., *DB Schema Editor*, 2008 <http://www.dbschemaeditor.com/OnlineDB.aspx>
- [7] Suraweera P., *An intelligent teaching system for database modelling*, University of Canterbury. 2002
- [8] Suraweera, P., Mitrovic, A.: KERMIT: a Constraint-based Tutor for Database Modeling. In: S. Cerri, G.

- Gouarderes, F. Paraguacu (eds.) Proc. ITS 2002, LNCS Vol. 2363 Springer-Verlag, (2002) 377–387
- [9] Chen, P. P., *The entity-relationship model--Toward a unified view of data*, ACM Transactions on Database Systems, 1(1), 9-36, 1976, Retrieved from <http://bit.csc.lsu.edu/~chen/pdf/erd-5-pages.pdf>
- [10] Object Management Group, *Unified Modelling Language*, 2007, <http://www.uml.org/>
- [11] Baranovskiy D., *Raphael JavaScript Library*, 2010, <http://raphaeljs.com/>
- [12] The World Wide Web Consortium (W3C), *Scalable Vector Graphics (SVG) 1.1*, 2011, <http://www.w3.org/TR/SVG11/>
- [13] The World Wide Web Consortium (W3C), *Vector Markup Language (VML)*, 1998, <http://www.w3.org/TR/NOTE-VML>
- [14] Bogdanović M., Stanimirović A., Davidović N., Stoimenov L., *The Development and Usage of a Relational Database Design Tool for Educational Purposes*, InSITE 2008, Varna, Bulgaria, , Informing Science Institute, pp. 251-258, ISSN: 1535-07-03, 2008 <http://proceedings.informingscience.org/InSITE2008/InSITE08p251-258Bodg505.pdf>
- [15] Bogdanović M., Davidović N., Stoimenov L., Stanimirović A., *#EER – alat za projektovanje relacionih baza podataka*, ETRAN 2007, Herceg Novi, 4-8. Jun, 2007, Društvo za ETRAN, ISBN: 978-86-80509-62-4, (in Serbian), 2007 <http://etran.etf.rs/etran2007/sekcije.htm>