Computers in Human Behavior 48 (2015) 273–289

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

# Simple-talking database development: Let the end-user design a relational schema by using simple words

Protogeros Nicolaos [a], Tzafilkou Katerina [b,*]

[a] University of Macedonia, Department of Accounting and Finance, 156 Egnatia Street, 54006 Thessaloniki, Greece
[b] University of Macedonia, Postgraduate Program in Information Systems, 156 Egnatia Street, 54006 Thessaloniki, Greece

ARTICLE INFO

*Article history:*
Available online 19 February 2015

*Keywords:*
Relational database schema design
Database-driven application development
User abstraction
End-user development
End-user behavior

ABSTRACT

Since every-day users are not aware of database concepts, it is almost impossible for them to design even the basic structure of their self-developed applications with existing low level database-developing prototype applications. Most of the existing prototype applications demand at least some basic knowledge of SQL syntax and/or database logic. Hence, the end-users have either to be trained to use such prototype applications or the prototype applications have to be designed in such a way as to support untrained end-users. In this paper we present an approach based on 'simple-talking' to abstract database terminology and logic from the application development process. In this approach we assist untrained users to progressively design the whole database structure of their applications. The idea behind 'simple-talking' approach is to let the user choose a verb that describes the relationship between entities, and use this verb to form questions for the various database items. Our approach follows the relational paradigm since it is the dominant database type and the most complicated for the end-user to understand. Based on a prototype application we have conducted experiments to evaluate the effectiveness of 'simple-talking' approach and reveal its usefulness for end-user development.



## 1. Introduction

End-User Development (EUD) is usually defined as a set of methods, techniques, and prototype applications that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artefact (Lieberman, Paternò, & Wulf, 2006). In other words, people who are not professional developers can use EUD prototype applications to create or modify software artefacts and complex data objects without significant knowledge of a programming language.

So far end-users mostly used EUD prototype applications for the web (EUDweb) to simply present information, create online forms, reports or surveys, since the deployment of complete database-driven web applications seemed impossible for non-professional developers. Today, interacting with databases has become commonplace in most end-user activities. Nowadays, modern computing advocates the idea of end-users using and creating databases in order to accomplish their daily tasks. This gives rise to the need for database design by the end-users. However, traditional database design is still heavy-weight, requiring technical expertise that end-users do not possess. Only a few articles address EUD issues in the context of database interaction. Early research in EUD is mainly focused on development and tailoring of individual performance prototype applications in single user work environments like e.g. computer aided design and excel sheets (e.g. Deng, Churcher, Abell, & McCallum, 2011). Most of these limited works suggest that end-users should act as co-designers and work with IT professionals (e.g. Fischer, Giaccardi, Ye, Sutcliffe, & Mehandjiev, 2004; Wulf, Pipek, & Won, 2008) or are targeted at the end-users' facilitation to build queries in order to manipulate, search and retrieve data from existed databases (e.g. Akiyama & Watanobe, 2012; Fan, Li, & Zhou, 2011; Li, Fan, Wu, Wang, & Feng, 2011; Wu, Li, & Zhou, 2010).

Almost every suggested approach for the end-users to develop a relational database structure demands at least some basic SQL knowledge or familiarity with database terminology (e.g. Borges & Macías, 2010; Brdjanin & Maric, 2013; Deng et al., 2011; Fraternali, Comai, Bozzon, & Toffetti, 2010). The empirical research in EUD indicates that end-users are unfamiliar with the database logic since it seems not to be close to their natural way of thinking (mental model) (Rode, Bhardwaj, Pérez-Qui~nones, Rosson, & Howarth, 2005; Rode & Rosson, 2003; Rode, Rosson, & Pérez-Quiñones, 2004, 2006). New approaches should be traced to assist the end-users design effective database schemas without the need

* Corresponding author. Tel.: +30 2310 891681.
*E-mail addresses:* proto@uom.edu.gr (P. Nicolaos), katerinatzaf@gmail.com (T. Katerina).

http://dx.doi.org/10.1016/j.chb.2015.02.002
0747-5632/

to be trained on database concepts and terminology. EUD prototype applications should follow the end-users' viewpoint instead of imposing them to follow the database logic as it is defined by professional developers (Scaffidi et al., 2008).

However, creating EUD prototypes that provide a decent user performance requires a deep understanding of end-user capabilities, limitations and needs. This paper focuses on human-centric database-driven web applications. The main research question is "Can non experienced and untrained end-users efficiently design relational schemes to develop database driven applications?". Based on this question, our essential concern is to render non experienced end-users capable of building a database driven application and efficiently design its relational schema without the need to be trained in database logic and terminology. The main objective of this research is to suggest and design a prototype user-centred approach to assist end-users build their own database applications (based on relational schema) without the need to know SQL syntax and/or database design theory.

Based on the recent research finding and monitoring of students interaction with tools for application design, we came up with the following idea: since user perception of things varies a lot among different personalities and culture, let the user choose the wording and verbs that best describe the relationships between the objects that need to be stored in the database and then use this wording in a wizard to guide her/him to provide the rest of the information (attributes, data types, etc.). Hence, we have designed and suggest propose in this work a prototype EUD approach named 'simple-talking'.

The 'simple-talking' approach assists the end-users to create tables, fields, correlations, integrity constrains, primary and foreign keys, define data types in a natural way abstracting him/her from these terms' semantics, resulting in the creation of a basic relational database schema. The 'simple-talking' approach gives the impression to the user that he/she is 'discussing' with the system in simple words while describing the ER model of his/her personal database application. Finally the system automatically generates the relational schema according to the user description.

We found no other approach in the literature review that can support the complete design of a relational schema by an end-user who is totally unfamiliar with the relational logic. The added value of this work is that it attempts to integrate a prototype user-centered approach in the EUD tools in order to finally render the end-users capable of effectively and efficiently build their own database-driven applications and unconsciously build the database relational schema. No other approach so far has attempted to follow the end-user way of thinking and replace the database logic and terminology to every-day words, so as to help the user design a database schema without the need to be trained. The 'simple-talking' approach attempts to combine behavioral research and software engineering to provide the HCI community with next generation and more 'natural' EUD tool environments.

To evaluate the effectiveness of the 'simple-talking' approach we designed a prototype EUD tool and conducted an experiment on a sample of end-users. The participants used the tool to design a prototype database relational model. After the task completeness the participants answered on certain questions regarding their perceived usefulness and ease of use of the system.

By analyzing the results we found that most of the participants performed well to build their own relational database schema since they achieved to construct most of the desired items, such as tables, fields and relationships. The 'simple-talking' approach generated automatically primary and foreign keys (based on the user descriptions). The experimental results seem encouraging and reveal that 'simple-talking' approach can form a new research direction to the EUD community. The results could also be used by the research community to study and analyze the end-user behavior and performance while interacting with EUD environments and especially when attempting to develop database applications. '*Let the end-user unconsciously design a relational schema by using simple words*' is our main motivation and we hope that our approach is a significant and innovative contribution to the fulfillment of this need. Our suggested approach can be used as the stepping stone to design user-centred database development environments that can fully support and integrate the relational model.

This paper is organized as follow. Section 2 presents some related work on the database design by end users and discusses its current limitations. Section 3 presents some basic literature findings about the end users' perception for the database logic so as to step on these to build the suggested approach. Section 4 presents the 'simple-talking' approach, describing and explaining most of its basic methodological parts. Section 5 describes our evaluation methodology and presents all the steps been taken to conduct the experiment to evaluate our approach. Section 6 provides with an overview of the prototype system design that was used in the experiment as the simple-talking EUD system and Section 7 presents the user story (exercise) that the participants were asked to achieve through the experiment, using the prototype system. In Section 9 we present our four research hypotheses explaining the reasons we 'constructed' them. In Section 10 there is some basic data analysis and the statistical experimental results, while in Section 11 we discuss these results providing the user with a clear image of the 'simple-talking' evaluation. Finally, in Section 12 we summarize the 'simple-talking' approach, underlining some result derived conclusions, discussing its implications, presenting and analyzing the 'simple-talking' limitations and suggesting some basic future work for the enhancement of the 'simple-talking' approach.

## 2. Related work

Multiple approaches have been developed to support the fast creation of custom database-driven web applications by removing the need to use a programming language. Examples of such approaches are model-driven (Brdjanin & Maric, 2013; Fraternali et al., 2010) and goal-driven approaches (Adi et al., 2008). However, the level of abstraction offered to the users by EUD model-driven or goal-driven prototype applications is not sufficient. As a matter of fact the abovementioned approaches refer to End-User Programming rather than to End-User Development. This implies that systems using these approaches are still not usable by end-user developers since they require basic (if not advanced) knowledge of the underlying modeling language and maybe some database design logic and relational schema terminology. The end-users have to be familiar with the design of Entity-Relational diagrams, tables' construction, relationships assignment and data validation in order to use them efficiently. For this reason, database researchers have recently suggested other approaches to address the 'end-user issue'. However, as we will see none of them fully addresses the development task of relational database–driven web applications from scratch.

Following we present some of these approaches existing in the state-f-the art EUD literature. We attempted to isolate and present only the database related EUD prototype applications and approaches, despite their deployment nature (desktop, web, mobile) since we first need to focus on the database development field and not on the more generic End-User Development for the Web (EUDWeb) area.

In their EUD research work, Rode et al. (2005, 2006) suggested and developed the PhpClick prototype application that provides the end-user with a comprehensive EUD environment to build