Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Evaluation of a flowchart-based EHR query system: A case study of RetroGuide ☆

Vojtech Huser [a,b,*], Scott P. Narus [c], Roberto A. Rocha [d]

[a] *Biomedical Informatics Research Center, Marshfield Clinic Research Foundation, 1000 North Oak Avenue, Marshfield, WI 54449, USA*
[b] *Morgridge Institute for Research, P.O. Box 7356, Madison, WI 53707, USA*
[c] *Department of Biomedical Informatics, University of Utah, 26 South 2000 East, Salt Lake City, UT 84112, USA*
[d] *Clinical Informatics Research & Development, Partners HealthCare System, 93 Worcester Street, Wellesley, MA 02481, USA*

ARTICLE INFO

Article history:
Received 5 November 2008
Available online 26 June 2009

Keywords:
Informatics
Biomedical informatics
Data warehouse
Evaluation
Mixed-method
Query system
RetroGuide
SQL

ABSTRACT

Provision of query systems which are intuitive for non-experts has been recognized as an important informatics challenge. We developed a prototype of a flowchart-based analytical framework called RetroGuide that enables non-experts to formulate query tasks using a step-based, patient-centered paradigm inspired by workflow technology. We present results of the evaluation of RetroGuide in comparison to Structured Query Language (SQL) in laboratory settings using a mixed method design. We asked 18 human subjects with limited database experience to solve query tasks in RetroGuide and SQL, and quantitatively compared their test scores. A follow-up questionnaire was designed to compare both technologies qualitatively and investigate RetroGuide technology acceptance. The quantitative comparison of test scores showed that the study subjects achieved significantly higher scores using the RetroGuide technology. Qualitative study results indicated that 94% of subjects preferred RetroGuide to SQL because RetroGuide was easier to learn, it better supported temporal tasks, and it seemed to be a more logical modeling paradigm. Additional qualitative evaluation results, based on a technology acceptance model, suggested that a fully developed RetroGuide-like technology would be well accepted by users. Our study is an example of a structure validation study of a prototype query system, results of which provided significant guidance in further development of a novel query paradigm for EHR data. We discuss the strengths and weakness of our study design and results, and their implication for future evaluations of query systems in general.

1. Introduction

Many healthcare organizations today maintain an enterprise data warehouse (EDW) with large volumes of clinical data [1,2]. These data represent a great opportunity for projects in quality improvement [3] or biomedical research [4]. EDWs, however, are very complex, and significant knowledge and experience are required for most query tasks [5]. Dorda et al. [6] and Chute [7] both indicate that user-friendly clinical query systems represent a considerable informatics challenge. Schubart's survey [5] of EDW clinical users and analytical staff showed that as many as 31% of the users with an EDW logon account reported that they never personally submitted a query to the EDW because of technological barriers such as necessary knowledge of the computer software, required training time, and complexity of the coding, financial or other data structures.

There are two fundamental ways of querying EDW data: *direct authorship of the query code* (the user constructs the query logic in a low-level query language) or use of a *query-building tool* (a specific query application assists the user in the query composition). Direct authorship of the query code is very similar to conventional programming and requires substantial expertise in a given query language, plus substantial knowledge of the underlying database schema [8]. Direct code authorship is often used for complex queries, the only restriction being the query language syntax. A non-expert EDW user usually collaborates with an expert analyst, knowledgeable of the EDW data structures and query technologies. Examples of query languages used to query clinical databases are: Structured Query Language (SQL), TimeLine SQL (TLSQL) [9], or AMAS language [6]. Query-building tools, on the other hand, are specifically designed for a non-expert user and offer a set of predesigned features which are easier to use then direct query code authorship. A classic example of a query-building tool is query design view within Microsoft Access. Examples of query-building tools for healthcare data are: (a) institution specific tools such

☆ The study was conducted while Vojtech Huser and Roberto Rocha were affiliated with the Department of Biomedical Informatics, University of Utah, and with Intermountain Healthcare, Salt Lake City, Utah.

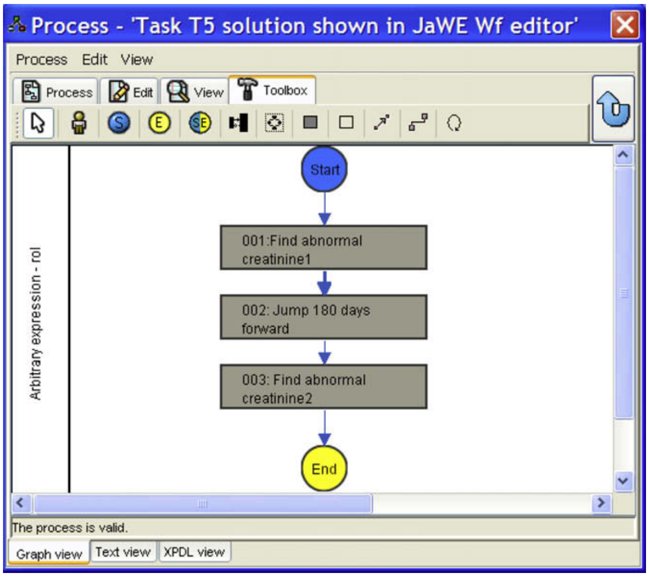
* Corresponding author. Address: University of Utah, and with Intermountain Healthcare, Department of Biomedical Informatics, Salt Lake City, UT, USA. Fax: +1 715 389 3808.

E-mail address: huser.vojtech@marshfieldclinic.org (V. Huser).

1532-0464/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jbi.2009.06.001

as: RPDR [10], STRIDE [11], SPIN [12]; and (b) publicly available tools such as: i2b2 Data Repository Cell [13]. A query building tool provides an additional query modeling layer (often involving a graphical metaphor) which eventually generates query code in one or a combination of several query languages. Although a query-building tool enables non-experts to execute queries unaided, it often limits the query expressiveness when compared to the direct code authorship. A common challenge of many query building tools is a case of a complex query which can be solved by an experienced analyst using the underlying low-level query technology (or a combination of several technologies), but it is not possible to author such query within the query building tool. This limitation can be caused by several factors: (1) limited tool's user interface; (2) the chosen graphical metaphor or the tool's native modeling paradigm can not support all necessary query criteria; (3) limited capability to combine interim solution layers within the tool (e.g., output of one query criterion is input for another criterion) or (4) the underlying low-level query language is too restrictive and can not be extended with user defined functions or combined with additional technologies within the tool.

We developed an analytical framework called RetroGuide [14,15] to address some of the query systems challenges mentioned above, and the focus of this paper is to present a mixed method evaluation of the RetroGuide prototype. RetroGuide is a suite of applications which enables a more user-friendly analysis of EDW data. RetroGuide uses, as graphical query metaphor, step-based flowchart models (called "scenarios") to represent queries. A RetroGuide scenario has two layers: a graphical *flowchart layer* and a hidden *code layer*. The flowchart layer (see Fig. 1 for an example) can be created and reviewed by users with limited expertise in database and query technology (e.g., champion clinicians or other non-expert requestors of EDW analyses such as administrative and management level healthcare personnel). The code layer is hidden behind the nodes and arrows of the flowchart and contains references to modular applications which can obtain EHR data or provide various analytical functions. RetroGuide query framework is extensible through addition of new modular applications, and the user can use scenario variables to combine related query criterions. The very close relationship between the scenario flowchart and the query execution engine goes beyond the traditional functionality of a query building tool and has many similarities to direct code authorship using a procedural and extensible query technology.


Process - 'Task T5 solution shown in JaWE Wf editor'
Process Edit View
Process
Edit
View
Toolbox
Arbitrary expression - rol
Start
001:Find abnormal creatinine1
002: Jump 180 days forward
003: Find abnormal creatinine2
End
Graph view
Text view
XPDL view

The process is valid.

Fig. 1. Viewing a RetroGuide scenario in a JaWE workflow editor. Referenced RetroGuide external applications can be viewed when double-clicking on a flowchart node. The scenario concurrently shows the RetroGuide solution to task question T5 in the evaluation study.

1.1. Lack of standards for evaluating query systems

As an introduction to our evaluation study design, we provide a brief overview of prior evaluations of related query systems. The findings of such review influenced our study design; however, it was not our goal to provide a generic query system evaluation methodology which addresses all possible challenges of such evaluation.

It is methodologically difficult to evaluate advanced data query systems and only a subset of previous publications about query systems includes an evaluation section [10,16–18]. The spectrum reflecting the degree of formal evaluation component in query systems publications would be: (1) no formal evaluation method presented (system features or methodology are descriptive only), (2) partial presentation of several example queries, with or without comparison to other query technologies, (3) complete single or multiple case studies (query and results) where the system is used to solve a concrete analytical problem, (4) presentation of system usage statistics demonstrating technology adoption by users, (5) qualitative stand-alone evaluation of the system with discussion of features distinguishing the system from previous similar efforts, and (6) comparative evaluation study using some qualitative measures to contrast the system against an existing technology.

Challenges to rigorous evaluation include the fact that innovative query technologies are often only evaluated in a prototype stage because many projects never reach widespread use where the technology would be refined to a user-friendly final product. The prototype status prevents researchers from conducting a proper laboratory-based evaluation. The prototype may contain a fully developed query language or engine, but lack a fully developed user-friendly query interface. Moreover, many of the products were primarily used within the originating institutions, i.e., use outside of these institutions would require substantial system adjustments; the systems are commonly not available for download and subsequent deployment at diverse sites. Finally, there are no standardized collections of queries ("test cases"), which would be regarded as representative of the analytical challenges of a given domain, but at the same time unbiased (system neutral). Query technologies from each unique domain focus on the specific and different challenges of that domain. For instance, within the healthcare domain, extending query technologies to better handle temporal reasoning is the special focus of many research-originated query systems [9].

Some of the biomedical query system evaluations which demonstrate the aforementioned methodological difficulties are ArchiMed [16] and the AMAS query language [6], DXtractor [17], Chronus [18] and the TimeLine Structured Query Language (TLSQL) [9], ACT/DB [8], and PROTEMPA [19]. Presenting several example queries and clinical case studies are the two most frequently used evaluation approaches. While example queries and case studies can be useful in understanding and demonstrating the new technology, they do not constitute a thorough evaluation.

TLSQL is currently the only technology with a formal quantitative comparative evaluation versus the structured query language (SQL), the most established query technology. Interestingly, there are no rigorous studies exploring the use of SQL by non-experts (e.g., clinicians or healthcare administrative personnel) and, in particular, their ability to solve a larger spectrum of query tasks. However, multiple qualitative reports do indicate that composing advanced queries in SQL requires substantial expertise [5,8,20].

Apart from looking at the technology's ability to model queries from a given corpus of problems, the evaluation of query systems