Published in final edited form as:
J Biomed Inform. 2010 February ; 43(1): 41–50. doi:10.1016/j.jbi.2009.06.001.

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

Vojtech Huser[a,b,*], **Scott P. Narus**[c], and **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

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.

Keywords

Biomedical informatics; Data warehouse; Evaluation; Mixed-method; Query system; RetroGuide; SQL

* Corresponding author. Tel: + 1 715 387 9140. Fax: + 1 715 389 3808. huser.vojtech@marshfieldclinic.org.
The study was conducted while Vojtech Huser and Roberto Rocha were affiliated with the Department Biomedical Informatics, University of Utah, and with Intermountain Healthcare, Salt Lake City, Utah.

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 pre-designed 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 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 Figure 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.

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 can be approached from a user's perception of the new technology. Research on information technology acceptance offers several models which combine inputs from the theory of reasoned action, the motivational model, the theory of planned behavior, the innovation diffusion theory, and the social cognitive theory [21]. In biomedical informatics, a technology acceptance model described by Davis [22,23] and Bagozzi [22,23] has been successfully applied in evaluations of several informatics resources [24-27]. However, a technology acceptance model has apparently not been applied to the evaluation of query systems.

The most recent technology acceptance theory has been formulated by Venkatesh et al. [21] and is known as the “Unified Theory of Acceptance and Use of Technology” (UTAUT). Venkatesh's model clearly defines determinants of user's *intention to use*, as well as *actual use* of a given resource. UTAUT offers validated constructs and survey instruments which can be used in system evaluation.

Outside the biomedical domain, similar methodology limitations in query systems evaluation can be observed [28]. Two types of evaluated systems are: (1) query languages with specific advanced capabilities [29,30], and (2) query systems using a visual metaphor to help the user in query formulation [31-33]. We have also reviewed evaluation strategies for a related field of natural language interfaces to databases [34-36]. The use of demonstrative examples is again the most common evaluation strategy, while the percentage of correctly generated queries, the percentage of unambiguous queries, and the query processing time are the most common quantitative measures. Murray et al. [33] describe a comparative, task-based evaluation of Kaleidoquery versus object query language using two groups of users (programmers and non-programmers).

1.2. RetroGuide

For better understanding of our RetroGuide evaluation study, we provide a brief description of RetroGuide. RetroGuide is based on workflow technology's [37] idea of executable flowcharts [14,15]. Thus, a query is viewed as a patient-centered, step-based process. The flowchart layer of a RetroGuide scenario represents these steps graphically and the execution of the flowchart is enabled by references to modular RetroGuide *external applications* (RGEAs) that provide specific data processing tasks, such as obtaining patient data, or performing data transformation and analysis. The RetroGuide suite of tools includes a basic starting set of these modular applications. This set can be further extended based on specific needs for additional analytical tasks. RetroGuide also utilizes the workflow technology concept of *variables* to transfer relevant data between individual scenario steps. RetroGuide uses the “XML Process Definition Language” (XPDL) as the underlying workflow definition language [38]. Currently, an open source workflow editor called JaWE [39] is used to model RetroGuide scenarios, but other XPDL-compliant editors could be used as well [40,41]. An example scenario viewed in JaWE is presented in the figure. A RetroGuide scenario consists of nodes which represent individual scenario steps. Each node may contain the execution of one or more external RGEAs. Arcs connecting the nodes represent the flow of logic and may contain a *transition condition* which further restricts the scenario logic, or implements branching or repetition logic.

Our underlying RetroGuide development aim is to test the feasibility of workflow technology for flowchart-based analysis of retrospective electronic health record (EHR) data. The current framework and developed software components represent a prototype implementation, and the evaluation study was designed to test the query modeling paradigm of this prototype. The key question of this evaluation study was to demonstrate performance and qualitative advantages of RetroGuide's flowchart-based query modeling approach over SQL on a group of non-expert

users. Due to utilization of workflow technology, we do not anticipate fundamental changes to this flowchart-based query modeling paradigm. The presentation of query results via RetroGuide's hierarchical reports (generated during RetroGuide scenario execution) was not evaluated in this study. We anticipate several future improvements to the reporting capabilities of the current RetroGuide prototype. All four phases of RetroGuide use (data pre-processing, scenario development, scenario execution, and reports review) are described in detail in previous publications [14,15,42]. This evaluation was performed on RetroGuide version 2.3, which was developed at Intermountain Healthcare and the University of Utah's Department of Biomedical Informatics as a PhD dissertation project. Extensions to RetroGuide (not reflected in this evaluation) were later developed at a second institution (Biomedical Informatics Research Center of Marshfield Clinic Research Foundation). This later work on RetroGuide is part of a larger project of integrating workflow technology within an EHR system [43]. Because RetroGuide is based on workflow technology and a standard process definition language, other institutions can reproduce its search functionality with any XPDL-compliant workflow suite. A workflow editor, a workflow engine and an EHR data repository are the key components of the RetroGuide architecture. Limited additional documentation is available at http://retroguideexpr.wiki.sourceforge.net.

1.3 Previous RetroGuide case-based evaluations

To test the feasibility and advantages of RetroGuide, a two-phase evaluation was designed. The first phase, described only briefly in the next paragraph, used case studies to demonstrate and test various system features. The second phase was the mixed method evaluation.

Phase one evaluation was methodologically similar to the aforementioned studies describing related query systems. Five RetroGuide case studies have been published: (1) analysis of female Hodgkin's lymphoma patients [44], (2) analysis of hypertension guideline adherence in diabetics [45], (3) analysis of glucose protocol performance in intensive care unit patients [46], (4) analyses of adverse drug events after use of narcotics and hepatitis C treatment [15], and (5) quality improvement analyses in osteoporosis and cholesterol control [47]. The results of the first evaluation phase were 7 fully-developed RetroGuide use cases created by the authors of the RetroGuide system (two use cases have not been published). Demonstration of the RetroGuide use cases to external reviewers (medical informatics experts) hypothesized the following RetroGuide advantages: (1) improved query understanding by analysis requestors, (2) easier query formulation offered by the step-based, single-patient methodology (building on the analogy to the manual chart review), and (3) enhanced drill-down capabilities facilitated by the hierarchical arrangement of RetroGuide result reports and RetroGuide's ability to produce query/scenario execution trace for individual patients.

2. Methods

2.1. Study Design

The first two hypotheses derived from the case-based evaluation (improved query understanding and easier query formulation) were the basis for the design of the second phase. Phase two evaluation used a group of 18 human subjects to compare the ability to solve query tasks using RetroGuide versus SQL.

The evaluation targeted informatics users with minimal to moderate experience with database query technologies. Due to RetroGuide's key developmental goal to lower the technological barriers for non-expert users analyzing EHR data, this specific group was deemed most appropriate for the evaluation. Friedman and Wyatt [48] define several types of evaluation studies which are performed depending on the maturity of the evaluated resource. The types of studies range from needs assessment and structure validation in the development stage to

usability tests and laboratory (or field) user effect studies in the deployment stage. Due to RetroGuide's prototype nature, we pursued a structure validation of RetroGuide's modeling paradigm. Friedman and Wyatt [49] provide the following description of this type of evaluation: “Structure validation study addresses the static form of the software and is usually performed after a first prototype has been developed. The focus is on the appropriateness of the algorithms, data structures and knowledge bases.” Our motivation for this structure validation was to guide future development of RetroGuide and to evaluate whether RetroGuide meets a subset of the original development requirements. Specifically, the requirements evaluated were: a user-friendly modeling tool, sufficient expressivity for medical problems, ability to iteratively progress from simpler to more complex problems, and the executable flowchart concept (see [14] for a complete list of the RetroGuide design requirements).

We chose a comparative study design (i.e., comparing RetroGuide to another analytical framework) because of its ability to better demonstrate relative RetroGuide advantages and disadvantages. SQL was chosen as a reference technology because it is directly available for use (unlike some research-based query systems), non-proprietary, and an internationally accepted standard. Das's TLSQL evaluation [18] also used SQL as the reference technology.

The evaluation study did not target the entire RetroGuide prototype implementation, but focused on how questions are modeled, since it represents the key factor in RetroGuide's analytical approach and is not likely to fundamentally change in future enhancements to the framework. The output portion of the framework (the generated reports and how answers are presented to the user) will be the focus of a future evaluation.

The study design was influenced by the prototype nature of the implementation and by the structure validation purpose of the study. Additional limiting factors (and their impact on the study) were: (1) reasonable requirement of subjects' time (impacted the number of the study tasks for the subject to solve and the choice of paper-based format rather than actual technology use), (2) limited subjects' experience with database querying and EDW data analysis (impacted the complexity of the study tasks), (3) subjects' clinical expertise (impacted the choice of tasks' clinical domains), and (4) our ability to enroll a sufficient number of subjects (impacted the subjects' enrollment criteria).

2.2. Study format

A total of 19 subjects were enrolled in the study (one subject dropped out later). Enrolled subjects were used as “proxy users” [49] for analyst and requestors (Friedman's definition of a proxy user is a user which sufficiently represents the resource's intended user in a situation where it is not possible or affordable to use the intended users directly). The enrollment criteria required subjects to have experience with analytical problems, databases and SQL, and a biomedical informatics background. The first criterion reflects the fact that the target user group (for the first RetroGuide release) was a specific subgroup of non-experts: champion clinicians or other champion requestors. The majority of such requestors at our institution have some prior database experience. The second criterion, limiting the subjects to existing or past biomedical informatics students, was intended to ensure that the enrolled subjects would have some aspects of a clinical-user type and, concomitantly, some aspects of an analyst-user type (combination of sufficient clinical and technical knowledge). Each subject received a paper study packet designed according to the test-based evaluation approach proposed by Friedman for structure validation studies.

The study had two primary aims: The first aim was to compare how well RetroGuide and SQL support solving analytical tasks in the healthcare domain. This aim was implemented as a quantitative paper study which used scores on a task-based test as a key evaluated measure. This quantitative study had two groups of questions. *Task questions* compared subjects' ability

to construct a solution to a given analytical task. *Choice questions* compared how well a subject understands and can modify pre-constructed solutions to a given analytical task.

The second aim was to compare the subjects' experiences with RetroGuide versus SQL. This aim was implemented as a qualitative paper study. The qualitative study included open-ended questions comparing the subject's experience with RetroGuide and SQL, and validated questions based on a most recent technology acceptance model formulated by Venkatesh et al. (UTAUT) [21]. UTAUT-based questions focused only on RetroGuide technology. The article appendix lists all tasks (T1-T9), choice (C1-C5) and UTAUT-based questions.

The study paper packet also included a five-question background questionnaire. The questions were based on the UTAUT-defined moderators (age, gender, past education, level of SQL experience) and the subject's source of SQL knowledge (formal SQL course or self-learning). These were collected in order to perform subanalyses of the study results.

2.2.1. Quantitative evaluation—The quantitative evaluation consisted of a total of 14 questions (9 task questions and 5 choice questions). Each subject was asked to answer the set of 14 questions twice, once using SQL and once using RetroGuide. The subjects were randomized so that half of them started with RetroGuide while the other half started with SQL. Subjects were instructed to have a 24-hour interval between the two approaches tested (wash-out period).

Each of the nine task questions (T1-T9) involved an analytical task to be solved, and a large blank space was available below each question for the subject to provide, in the SQL case, a narrative, SQL-based pseudo-code solution, or, in the RetroGuide case, a drawing of a flowchart solution annotated with the employed RetroGuide external applications. For example, question T4 was: “Find all patients who had at least 2 creatinine lab results flagged as too high.”

Subjects were instructed to focus on the overall ability to solve a given task and describe fundamental steps necessary for solving it. Minor technology syntax errors were specifically stated as outside the focus of the task questions. For scoring purposes, each task question had, apart from several general query correctness criteria (for RetroGuide and SQL), an itemized set of crucial solution steps. The presence of those crucial steps in a subject's solution was evaluated on a percentage range (0-100%), which was then translated into 3 possible question scores. For example, the list for RetroGuide's solution to question T5 (“Find all patients who had at least 2 creatinine lab results too high but they must be at least 180 days [or more] apart from each other”) included: (1) proper identification of two, abnormal creatinine results, and (2) correct implementation of the 180-days-apart criterion. Partial fulfillment of those crucial steps was also considered. Each task question would be later scored 0, 0.5, or 1 point. A correct solution received 1 point, a partial solution containing at least 50% of the necessary steps received 0.5 points, and an incorrect or ‘blank’ solution received 0 points. The final, three-grade scoring granularity (0, 0.5, or 1 point) for each question simplified the overall test scoring. Two reviewers (RAR and SN) were used to assign scores for each task question. The reviewers were senior-level informaticists, each with over 15 years of experience in medical data analysis and decision support.

Subjects were instructed to use no more than 45 minutes to solve all task questions in a given technology (RetroGuide or SQL). Use of the time limit ensured that the study also factored into the comparison the time required for solving a problem in a given technology. The technology which supported faster solution creation or review had a higher chance of achieving higher average test scores.

The five choice questions (C1-C5) introduced a solution to a given analytical problem (1), and then presented a slightly extended problem (2) together with three possible solutions (marked "a", "b", or "c"). The subjects were asked to choose the correct option. Option "c" in each question was "none of the above" and was designed to be the correct answer on one of the five questions for each tested approach to broaden the number of options [50]. Choice questions were later scored 1 point for a correct solution and 0 points for an incorrect or missing one. Similarly to the task questions section, a 15-minute time limit was enforced for the completion of choice questions (separately for each tested technology).

A total test score for RetroGuide answers and for SQL answers was calculated. A paired t-test was used to compare the mean RetroGuide and SQL scores. The null, two-sided hypothesis was that there is no difference in mean scores of RetroGuide versus SQL technology. Each of the two subsections (choice questions and task questions) was also analyzed separately using the same statistical approach. To assess the agreement of the two human reviewers (for the task questions), we calculated a Cohen's kappa statistics and performed three subanalyses of the task questions score differences: separately for each reviewer and as a combined average score.

2.2.2. Qualitative evaluation—The second part of the study focused on qualitatively comparing the two approaches (SQL vs. RetroGuide) using 13 follow-up questions (F1-F13) divided into two major sections A and B. Section A focused on comparison of SQL and RetroGuide technologies, and also included one question about a generic analytical technology. Section B questions targeted RetroGuide technology only.

The first three questions (F1-F3) of section A were open-ended questions. Question F1 asked what approach was preferred by the subject and why. Questions F2 and F3 asked the subject to state all disadvantages found with SQL and RetroGuide, respectively. Section A concluded with one question (F4) about the quality of the instructions provided for both parts of the quantitative study, and a final question (F5) where 9 features of a generic analytical tool were ranked by the subject according to their perception of importance.

Section B used Likert scale questions (F6-F13) to assess RetroGuide, which were based on previously validated constructs of the UTAUT model. UTAUT unifies several previous technology acceptance models. In biomedical informatics, the technology acceptance model described by David and Bagozzi [22,23], one of the UTAUT predecessors, has been used in several informatics technology evaluations [24-27]. UTAUT defines four direct *determinants* (performance expectancy, effort expectancy, social influence, and facilitating conditions) of user *intention to use* a technology and also looks at later *actual use* of the technology. It identifies four *moderating factors* which affect the above four determinants (gender, age, experience, and voluntariness of use).

Section B investigated 3 UTAUT constructs by using at least 2 questions per construct: performance expectancy (questions F6, F9, F12), effort expectancy (questions F7, F10, F13), and behavioral intention (questions F8, F11). UTAUT provides a validated 5-point Likert scale ranging from 1 (=strongly disagree) to 5 (=strongly agree). For example, question F6 investigating performance expectancy was: "I find RetroGuide useful for solving analytical problems;" question F10 investigating effort expectancy was: "It is easy for me to use RetroGuide to create analytical models;" and question F8 investigating behavioral intention was: "Having RetroGuide as an available option in my analytical job – I intend to use RetroGuide." Knowledge of the user's assessment of those three constructs for RetroGuide technology, according to the UTAUT model, enables prediction of later actual use of RetroGuide-like technology. This prediction is, however, limited by the fact that none of the qualitative evaluation's questions attempted to assess UTAUT's social influence and facilitating

conditions predictors. Given the structure validation setting of the study and the prototype nature of the tested resource, it would not have been meaningful to include these two predictors.

2.3. Study procedure

Subjects were enrolled using email and a personal 5-minute interview describing the goals and structure of the study. Enrolled subjects met with the investigator (VH) for 45 minutes where standard training on RetroGuide technology was presented and they were given the study packet. The packet contained specific instructions for the subject to perform the study on his/her own schedule, including the 24-hour break between the compared approaches and the time-limit restrictions for each section. The packet contained 3 additional items: (1) summary of the RetroGuide technology training; (2) simplified list of necessary RetroGuide external applications; and (3) sample data sheet demonstrating the storage structure of the EHR data, which applied to both compared technologies (SQL and RetroGuide). Training for SQL technology was not provided since a minimum of a basic knowledge of SQL was an enrollment criterion for subjects, and the study assumed that a basic knowledge is more than equivalent to the 45-minute RetroGuide training.

After return of all study packets, the qualitative section was then scored and analyzed using R statistical package [51]. Manual content analysis of the qualitative section was done using categorization, summarization and tabulation techniques on the collected textual data [52]. The reliability of the tested UTAUT constructs in the qualitative section B was analyzed using the same statistical package.

3. Results

Characteristics of the 18 subjects are shown in Table 1. The majority of the subjects were female (72.2%) and of age category 31-35 (33.3%). Intermediate knowledge of SQL was most prevalent (44.4%), and zero subjects reported no SQL knowledge or expert SQL knowledge, satisfying the aim of the enrollment criteria.

3.1. Quantitative evaluation

The mean total RetroGuide score was 11.1 compared with the mean SQL score of 6.3. The mean difference of 4.8 ± 1.8 was statistically significant using paired t-test ($p < 0.001$, 95% confidence interval 3.4-5.4). Similarly, significant results were found when looking at subscores of task questions (T1-T9) only, as well as subscores of choice questions (C1-C5) only (see Table 2 for complete results). Task questions T1-T9 were scored by two different reviewers. The total score results presented above and in Table 2 use the average score from both reviewers. A separate analysis of the scores for each reviewer (task questions only) agreed with the above presented average scores results ($p < 0.001$ for both reviewers, data shown in Table 3).

The Cohen's kappa statistic was used to determine the degree of agreement of the two reviewers on the total pool of 234 task question score pairs. The result (kappa=0.53) indicates moderate agreement [53].

Linear regression was used to determine whether score difference could be predicted by any of the subject characteristics such as gender, age, SQL experience, or SQL experience source. No linear regression model could predict the score difference (adjusted R-squared < 0.1) and none of the factors were statistically significant. A two-sample t-test showed no statistical difference in test score differences between the group which started with the SQL technology versus the group which started with the RetroGuide technology.

3.2. Qualitative evaluation

According to results from question F1 (“Which technology do you prefer and why?”), 94.4% (17 subjects) preferred RetroGuide to SQL. Analysis of the qualitative comments was performed and several categories were identified. The leading categories were “easy to learn/use/understand” (9 subjects), followed by “temporal modeling capabilities” (6 subjects), and “more intuitive/natural/logical” (4 subjects). One subject found both technologies equivalent. No subject preferred SQL. Such strong preference for RetroGuide, in fact, represents probably the most important overall result of the evaluation study.

Examining answers to the question on SQL's difficulties (question F2), the leading category of comments were “must know exact syntax/be expert” (7 subjects), “difficult to use” (6 subjects), and “insufficient support for temporal criteria” (5 subjects). The leading categories for RetroGuide's difficulties (question F3) were “need to know function of various applications and new terminology” (4 subjects), “none” (3 subjects), “hard to understand what data user gets back” (2 subjects), and “can be slow for queries involving a larger population” (2 subjects).

For the two questions which asked about clarity of study instructions, using a 5-point scale of very poor (1) to excellent (5), the SQL mean result was 4.06 and the RetroGuide mean was 4.24. Those scores indicate that the study instructions for both methods were sufficiently clear. Finally, for question F5 where the importance of several features of a generic analytical tool was evaluated, the results are shown in Table 4. The table lists overall rank for 9 prelisted features and one subject-added feature. The two most important features were “intuitive model understood by non-experts” (rank 2.22) and “short training time” (rank 3.61). These were followed by five other factors with similar average rank, ranging from 5.05 to 5.56 (“graphical representation,” “facilitation of collaboration,” “short query time,” “based on established technology,” and “direct access to data as physically stored”).

For the 3 UTAUT constructs investigated (questions F6-F13), Table 5 lists Likert scale mean and standard deviations for each question. It also shows Cronbach's alpha reliability statistics for each concept. All three surveyed constructs of performance expectancy, effort expectancy, and behavior intention exhibit reliability within the recommended range of > 0.70 [54]. Employing the UTAUT prediction model, the achieved mean scores indicate that if the RetroGuide technology would be developed from the current prototype into a full software product, it most likely would be well accepted by future users (all scores are well above the middle neutral score of 3, which indicates a favorable effect on the final measure of actual technology use).

4. Discussion

4.1 Discussion of results

The quantitative and qualitative results of our study indicate that RetroGuide is a better and preferred technology then SQL for non-expert users. By measuring scores on a test (consisting of query tasks) we were able to qualitatively show a significant difference between RetroGuide and SQL technologies. There are two evaluation studies in the biomedical informatics domain which have a very similar evaluation design and also use a task-based approach and score comparison methodology [55,56]. Qualitative results show strong user's preference of RetroGuide (95%) over SQL technology, reveal its key advantages and hint to possible improvements for future versions of RetroGuide – which is in agreement with the structure validation and development guidance purpose of our study.

The study showed three main qualitative advantages of RetroGuide over SQL: RetroGuide was easier to learn, better supporting temporal query tasks, and it was perceived to be a more intuitive modeling paradigm. We hypothesize that RetroGuide's analogy to step-based manual

chart review and other features related to this analogy (e.g., single-patient processing, or the concept of the current position in the record) are largely responsible for these three advantages. In terms of RetroGuide disadvantages, two possible groups can be discussed: (a) disadvantages identified by the study subjects during our evaluation, and (b) disadvantages identified by the creators of RetroGuide or other informatics reviewers. The following three RetroGuide disadvantage themes were most frequently stated by the study subjects (in response to question F3): (1) need to know what individual RGEAs do; (2) unclear query output; and (3) possible slow query performance on large cohorts. Two additional expert-suggested RetroGuide disadvantages were: (1) readability of flowchart of very large scenarios (involving more then 40 nodes), and (2) ability to search for multiple instances of given event patterns (e.g., pregnancy followed by a deep vein thrombosis) without requiring a loop flowchart structure.

The advantage of our evaluation over similar system evaluations (perhaps with the exception of Das's study) is that it compares RetroGuide's developed framework to another analytical formalism (SQL). Additionally, the employed measures consider the user's experience with the technology. Das's study used query time as a comparison measure. Given that a technically perfect technology may not be the most convenient framework to work with for a non-expert requestor or analyst, factoring in a user's experience with the technology may provide a more useful indicator of the overall quality of the analytical technology. Our study is a query system evaluation that uses the UTAUT technology acceptance model. Structure validation studies are an important part of developing any informatics resource, and designs and results of these studies are often underreported in literature. Our report is a contribution to help fill this gap.

A final observation made during this evaluation is the importance of the underlying technological difference between SQL and RetroGuide technologies, which impacts the way a solution to a query problem is constructed. Whereas SQL is a declarative programming language, RetroGuide scenario resembles a procedural programming language. In a declarative language, the user is using a fixed set of language-supported constructs (e.g., SELECT, WHERE, JOIN) to specify the desired goal, and the underlying query engine takes care of fulfilling the request algorithmically. In other words, the SQL statement reflects what the result should be, rather then how to obtain it. A declarative language needs a complex, underlying query processing engine to translate the declarative model into actual result-producing computer steps. The purely declarative nature of SQL led to the development of procedural extensions to SQL, such as PL/SQL or Transact-SQL. Query-building tools that rely primarily on SQL technology will likely expose this declarative way of problem-solving. Our qualitative results indicate that this declarative modeling might not be intuitive to non-experts. In contrast, a solution written in a procedural (or imperative) language defines a sequence of computer commands, authored directly by the user, which ultimately produces the results. The user's strong preference for RetroGuide indicates that the procedural method of solving tasks, wrapped into a graphical interface, is more intuitive for non-experts because it resembles how a human might solve the problem as a step-based, manual chart review.

4.2 Study limitations

A limitation of the study is that the results are valid for the specific and limited set of tasks that were analyzed. Although the set of analytical problems used was meant to be realistic and was reviewed by a panel of experts (researchers with experience and formal training in informatics and health services research), it was not realistic to attempt to cover all possible analytical tasks. The choice of tasks, their total number, and their complexity was, in fact, significantly limited by the study time requirements and by the clinical and analytical background knowledge of the targeted subject population. Furthermore, there was a special emphasis on tasks, which included temporal conditions, since temporal considerations are frequent and important in medicine, but current database and query technologies often lack sufficient

support for them [6]. An initial, larger set of 21 possible problems (for task and choice questions) was reduced to only 14 final questions using a panel of two study researchers. The smaller set was then presented to five medical informatics experts. Two experts provided suggestions which were later incorporated into the final set.

We did not find a suitable and established corpus of analytical tasks that we could have relied upon. Each new analytical technology focuses on different aspects of the data query problem and no scientific effort has been made in the past to implement the same set of problems with multiple related technologies and compare performance along multiple axes (e.g., query time, training and expertise required, result presentation, or overall technology user-friendliness). We did produce RetroGuide solutions to clearly defined problems in publications presenting related systems, but many of those problems were too complex or required substantial clinical or background expertise to be used in a fairly time-constrained study.

A second study limitation is its focus on users with low to intermediate levels of SQL knowledge, and whether requiring some SQL knowledge but not expert level is appropriate. As for the requirement of some SQL knowledge, we restate our focus on champion requestors, making the study results applicable only to a subset of non-expert users. We also wanted to evaluate RetroGuide against an existing and proven technology. SQL was selected because of its availability and general user experience, but only limited SQL knowledge was required of the participants. Regarding the exclusion of SQL experts, an important aim of the RetroGuide project was to offer a tool that could lower the barrier to analysis of data for novice users, i.e., novice in terms of limited SQL expertise and understanding the database schemas. Expert users of SQL most likely do not need support in solving complex query tasks. Thus, given the specific group of users that would benefit from RetroGuide, the imposed subject enrollment limitations were considered very important and acceptable.

Finally, the use of a paper-based evaluation format, rather than actual software tools for authoring and executing SQL queries, or a workflow process editor and engine for RetroGuide scenarios, may have also impacted our results. The ability of subjects to use advanced features of a SQL editing environment, such as code completion, color coding, or debugging, would likely produce different results; however, its impact on the qualitative results is less obvious since most errors in this evaluation were due to an inability to think of a proper SQL statement or overall strategy for a task, details most SQL editing environments do not address. The impact of RetroGuide's graphical user interface on query creation is unknown at this point and will be a subject of later evaluations. Our choice of a paper-based format was largely determined by the structure validation study format, but also time constraints for subjects to complete the study.

4.3 Lessons learned and recommendations

During our study several important challenges and opportunities were identified. These are summarized in the following points:

1. A structure validation study during an early stage of development of a novel query system can be very instrumental in guiding subsequent development. User-friendly query systems represent a considerable development challenge and evaluation efforts involving early prototypes provide very valuable feedback. However, the tendency to always apply very strict evaluation criteria may result in fewer research projects attempting to take advantage of this opportunity.
2. Finding optimal measures for quantitative comparison of query technologies is difficult. Measuring query execution time (as used in the comparison of TLSQL and SQL [18]) is relevant to a user who needs immediate results. However, the time required to author correct query statements may be a much more important factor.

Several other measurements, such as training time, necessary knowledge about healthcare data structures, or complexity of the testing corpus of tasks are also important. In an optimal evaluation, multiple measures should be used. Our solution was to combine all these factors into a score measure on a task-oriented test, which was then presented to a specific group of users within a limited time period.

3. Establishing comparable scoring instructions for diverse technologies and the assignment of scores to subject's solutions was very challenging. In non-prototype settings, this can be solved by an automated comparison of corresponding query outputs and use of discrete scoring (pass/fail), rather than a continuous scoring system. Increasing the number of solved tasks, if the study settings allow, is another way of optimizing the evaluation design.

5. Conclusion

This resource structure validation study compared RetroGuide with SQL-based tools using a sample of non-expert users. Using the RetroGuide approach, the subjects achieved significantly higher scores in solving analytical tasks, and also scored higher in tasks which required understanding of given analytical solutions. The qualitative part of the study demonstrated that most users preferred RetroGuide to SQL because RetroGuide was easier to learn, it better supported temporal tasks, and it was perceived to be a more logical modeling paradigm. Using the UTAUT technology acceptance prediction model, the study results suggest that a fully developed, RetroGuide-like technology likely would be well accepted by users.

The study results are crucial in our decisions about further enhancement and development of the RetroGuide framework. The future plan is to produce a more robust RetroGuide implementation which would be mature enough to undergo a more rigorous, laboratory, user-effect evaluation. The research community should consider establishing a standardized corpus of healthcare query tasks which would improve validity of comparative evaluations of different query systems.

Acknowledgments

The authors would like to thank Intermountain Healthcare for supporting VH during his PhD project (RetroGuide development and evaluation). VH is also supported by the Morgridge Institute for Research. We thank the writing center team of Marshfield Clinic Research Foundation for assistance with manuscript preparation.

Appendix

Quantitative task questions (T1-T9)

T1: Find all patients who ever had in their EHR any record of a previous “comprehensive eye exam” report.

T2: Find all patients who had blood creatinine level measured at least once.

T3: Find all patients whoever had blood creatinine lab results which were flagged as being too high.

T4: Find all patients who had at least 2 creatinine lab results flagged as too high.

T5: Find all patients who had at least 2 creatinine lab result too high but they must be at least 180 days (or more) apart.

T6: Find all patients who have diabetes but no record of hypertension diagnosis.

T7: Find all patients who have record of both diabetes and hypertension diagnosis in their EHR (the temporal order does not matter).

T8: Find all patients who were first diagnosed with diabetes and their diagnosis of hypertension came after their diabetes (certain temporal order of conditions enforced).

T9: Find all patients with diabetes, who have at least 2 systolic blood pressure measurements over 130 mmHg after they became diabetic. However, 2 additional restrictions apply to the question. First, do not consider any blood pressure measurement in the initial treatment period of 24 months after establishing the diagnosis of diabetes. Second, the 2 elevated systolic blood pressure measurements over 130 must be at least 11 months apart from each other.

Quantitative choice questions

C1: LDL Cholesterol test

Problem 1: Find patients who ever had LDL-cholesterol over 130 mg/dl.

Problem 2: Find all patients whose latest LDL cholesterol in year 2005 was over 130 mg/dl.

C2: Fracture in women

Problem 1: Find all patients who had a fracture.

Problem 2: Find all patients who had a fracture at age 66.

C3: Hypertension prior diabetes

Problem 1: Find all patients who have both conditions – diabetes and also hypertension. (temporal order does not matter).

Problem 2: Find all patients who have both conditions but they had diagnosis of hypertension first and after that became diabetic (specific order enforced).

C4: Count number of fracture episodes

Problem 1: Count how many fracture episodes each patient had.

Problem 2: Count the fracture episodes each patient had. But after any given fracture episode, do not count any follow-up fracture visit within 90 days.

C5: Adverse drug event detection

Problem 1: Find all patients who were given narcotic antidote naloxone and within 6 hours from this naloxone administration were transferred to ICU. If naloxone is given multiple times, consider only the first such episode.

Problem 2: Find all patients who experienced the above adverse drug event (naloxone and transfer within 6 hours) and also had a record of sleep apnea ICD diagnosis prior to this adverse drug event (i.e.: the searched sequence is: apnea, naloxone and transfer).

UTAUT-based qualitative study questions (F6-F13)

Each question was rated on a scale ranging from 1 to 5, where 1 = strongly disagree and 5 = strongly agree. Reliability scores for each construct are shown in parentheses.

PE: Performance expectancy (alpha = 0.871)

PE1: I find RetroGuide useful for solving analytical problems/questions.

PE2: Using RetroGuide enables me to accomplish analytical tasks more quickly.

PE3: Using RetroGuide increases my productivity.

EE: Effort expectancy (alpha = 0.849)

EE1: I find RetroGuide easy to use.

EE2: It is easy for me to use RetroGuide to create analytical models.

EE3: Learning to use RetroGuide is easy for me.

BI: Behavioral intention (alpha = 0.752)

BI1: Having RetroGuide as an available option in my analytical job - I intend to use RetroGuide.

BI2: Having RetroGuide as an available option in my analytical job - I predict that I would use RetroGuide.

References

1. Edward, FE.; Carl, EM.; Laura, ED. Data warehousing in an integrated health system: building the business case. Proceedings of the 1st ACM international workshop on Data warehousing and OLAP; Washington, D.C., United States. 1998.
2. del Hoyo E, Lees D. The use of data warehouses in the healthcare sector. Health Informatics Journal 2002;8(1):43–6.
3. Berwick DM. Continuous improvement as an ideal in health care. N Engl J Med 1989;320(1):53–6. [PubMed: 2909878]
4. De Clercq E, Van Casteren V, Jonckheer P, Burggraeve P, Lafontaine MF, Vandenberghe H, et al. Research networks: can we use data from GPs' electronic health records? Stud Health Technol Inform 2006;124:181–6. [PubMed: 17108523]
5. Schubart JR, Einbinder JS. Evaluation of a data warehouse in an academic health sciences center. Int J Med Inform 2000;60(3):319–33. [PubMed: 11137474]
6. Dorda W, Gall W, Duftschmid G. Clinical data retrieval: 25 years of temporal query management at the University of Vienna Medical School. Methods Inf Med 2002;41(2):89–97. [PubMed: 12061129]
7. Chute CG. Clinical data retrieval and analysis. I've seen a case like that before. Ann N Y Acad Sci 1992;670:133–40. [PubMed: 1309082]
8. Nadkarni PM, Brandt C. Data extraction and ad hoc query of an entity-attribute-value database. J Am Med Inform Assoc 1998;5(6):511–27. [PubMed: 9824799]
9. Das AK, Musen MA. A comparison of the temporal expressiveness of three database query methods. Proc Annu Symp Comput Appl Med Care 1995:331–7. [PubMed: 8563296]
10. Murphy SN, Gainer V, Chueh HC. A visual interface designed for novice users to find research patient cohorts in a large biomedical database. AMIA Annu Symp Proc 2003;2003:489–493. [PubMed: 14728221]
11. STRIDE tool. Available from: http://clinicalinformatics.stanford.edu/STRIDE
12. McDonald CJ, Dexter P, Schadow G, Chueh HC, Abernathy G, Hook J, et al. SPIN query tools for de-identified research on a humongous database. AMIA Annu Symp Proc 2005;2005:515–519. [PubMed: 16779093]
13. Murphy SN, Mendis M, Hackett K, Kuttan R, Pan W, Phillips LC, et al. Architecture of the open-source clinical research chart from Informatics for Integrating Biology and the Bedside. AMIA Annu Symp Proc 2007 Oct 11;:548–52. [PubMed: 18693896]

14. Huser, V.; Rocha, RA.; James, B. Use of workflow technology tools to analyze medical data. 19th IEEE International Symposium on Computer-Based Medical Systems Proceedings; Salt Lake City, UT. 2006. p. 455-60.
15. Huser, V.; Rocha, RA. Analyzing medical data from multi-hospital healthcare information system using graphical flowchart models. Proceedings of the Biomedical Informatics and Cybernetics Symposium; 2007. p. 314-320.
16. Dorda W, Wrba T, Duftschmid G, Sachs P, Gall W, Rehnelt C, et al. ArchiMed: a medical information and retrieval system. Methods Inf Med 1999;38(1):16–24. [PubMed: 10339959]
17. Nigrin DJ, Kohane IS. Temporal expressiveness in querying a time-stamp-based clinical database. J Am Med Inform Assoc 2000;7(2):152–63. [PubMed: 10730599]
18. Das AK, Musen MA. A temporal query system for protocol-directed decision support. Methods Inf Med 1994;33(4):358–70. [PubMed: 7799812]
19. Post AR, Harrison JH Jr. PROTEMPA: a method for specifying and identifying temporal sequences in retrospective data for patient selection. J Am Med Inform Assoc 2007;14(5):674–83. [PubMed: 17600103]
20. Schubart JR, Einbinder JS. Evaluation of a data warehouse in an academic health sciences center. Proc AMIA Symp 1999:614–8. [PubMed: 10566432]
21. Venkatesh V, Morris MG, Davis GB, Davis FD. User acceptance of information technology: toward a unified view. MIS Quarterly 2003;27(3):425–78.
22. Davis FD. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 1989;13(3):319–40.
23. Bagozzi RP, Davis FD, Warshaw PR. Development and test of a theory of technological learning and usage. Human Relations 1992;45(7):659–86.
24. Hulse NC, Del Fiol G, Rocha RA. Modeling end-users' acceptance of a knowledge authoring tool. Methods Inf Med 2006;45(5):528–35. [PubMed: 17019507]
25. Chismar WG, Wiley-Patton S. Test of the technology acceptance model for the internet in pediatrics. Proc AMIA Symp 2002:155–9. [PubMed: 12463806]
26. Wilson EV, Lankton NK. Modeling patients' acceptance of provider-delivered e-health. J Am Med Inform Assoc 2004;11(4):241–8. [PubMed: 15064290]
27. Hu PJ, Chau PY, Sheng OR, Tam KY. Examining the technology acceptance model using physician acceptance of telemedicine technology. J Manag Inf Sys 1999;16(2):91–112.
28. Catarci T. What happened when database researchers met usability. Information Systems 2000;25 (3):177–212.
29. Mason, T.; Wang, L.; Lawrence, R. AutoJoin: providing freedom from specifying joins. 7th International Conference on Enterprise Information Systems – Human-Computer Interaction Track; Miami, FL. 2005. p. 31-8.
30. Mason, T.; Lawrence, R. INFER: a relational query language without the complexity of SQL. ACM CIKM (14th ACM International Conference on Information and Knowledge Management); Bremen, Germany. 2005. p. 241-2.
31. Batini, C.; Catarci, T.; Costabile, M.; Levialdi, S. Visual query systems: a taxonomy; VDB. 1991. p. 153-68.Available from: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=6FE6592CC79AD84826ABE42F374A38A1?doi=10.1.1.45.8882&rep=rep1&type=pdf
32. Chen PK, Chen GD, Liu BJ. HVQS: the hierarchical visual query system for databases. J Vis Lang Comput 2000;11(1):1–26.
33. Murray NS, Paton NW, Goble CA, Bryce J. Kaleidoquery - A flow-based visual language and its evaluation. J Vis Lang Comput 2000;11(2):151–89.
34. Androutsopoulos I, Ritchie GD, Thanisch P. Natural language interfaces to databases - an introduction. Nat Lang Eng 1995;(1):29–81.
35. Nelken, R.; Francez, N. Querying temporal databases using controlled natural language. Proceedings of the 18th Conference on Computational Linguistics - Volume 2; Saarbrücken, Germany. 2000. p. 1076-80.

36. Androutsopoulos, I.; Ritchie, GD.; Thanisch, P. Experience using TSQL2 in a natural language interface. In: Clifford, J.; Tuzhilin, A., editors. Recent Advances in Temporal Databases (Proceedings of the International Workshop on Temporal Databases, Zurich, Switzerland), Workshops in Computing series. Springer-Verlag; 1995. p. 113-32.
37. Workflow Management Coalition. Terminology and glossary (WFMC-TC-1011). Available from: http://healthcareworkflow.files.wordpress.com/2008/10/wfmc-tc-1011_term_glossary_v3.pdf
38. Workflow Management Coalition. XML process definition language (XPDL). Available from: http://www.wfmc.org/xpdl.html
39. JaWE. Java workflow editor (XPDL). Available from: http://jawe.objectweb.org
40. Cape Visions™. XPDL Extension for Microsoft Visio. Available from http://www.capevisions.com/tech/wxml.shtml
41. Tibco Business Studio. Available from http://www.tibco.com/devnet/business_studio/default.jsp
42. Huser, V. PhD Dissertation. Salt Lake city: University of Utah; 2008. Analyzing biomedical datasets using executable graphical models. Available at http://content.lib.utah.edu/u?/us-etd2,61096
43. Healthcare Workflow (project blog). Available at http://healthcareworkflow.wordpress.com
44. Huser, V.; Rocha, RA.; Huser, M. Conducting time series analysis on large data sets: a case study with lymphoma. In: Kuhn, KA.; Warren, JR.; Leong, TY., editors. Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems; Amsterdam. 2007. Available from: http://content.lib.utah.edu/cgi-bin/showfile.exe?CISOROOT=/ir-main&CISOPTR=13383&filename=2714.pdf
45. Huser, V. Running decision support logic retrospectively to determine guideline adherence: a case study with diabetes. Spring AMIA2007 symposium; Available from: http://content.lib.utah.edu/cgi-bin/showfile.exe?CISOROOT=/ir-main&CISOPTR=12786&filename=2156.pdf
46. Huser, V.; Rocha, RA. Retrospective analysis of the electronic health record of patients enrolled in a computerized glucose management protocol. Twentieth IEEEInternational Symposium on Computer-Based Medical Systems (CBMS'07); 2007. p. 503-8.Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4262698
47. Huser, V.; Rocha, RA. Graphical modeling of HEDIS quality measures and prototyping of related decision support rules to accelerate improvement; AMIA Annu Symp Proc. 2007. p. 986Available from: http://content.lib.utah.edu/cgi-bin/showfile.exe?CISOROOT=/ir-main&CISOPTR=12995&filename=2365.pdf
48. Friedman, CP.; Wyatt, J. Evaluation methods in biomedical informatics. 2nd. New York: Springer; 2006.
49. Friedman, CP.; Wyatt, J. Evaluation methods in biomedical informatics. 2nd. New York: Springer; 2006. Chapter 3: Determining what to study; p. 48-84.
50. Good practice guide in question and test design. Available from http://www.pass-it.org.uk/resources/031112-goodpracticeguide-hw.pdf
51. R Development Core Team. R: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing; 2006.
52. Lacey, A.; DLuff, D. Qualitative research and data analysis. University of Nottingham; UK (Trent Focus Group): 2004.
53. Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics 1977;33(1):159–74. [PubMed: 843571]
54. Nunnally, JC. Pyschometric theory. 2nd. New York: McGraw-Hill; 1978.
55. Campbell R, Ash J. Comparing bedside information tools: a user-centered, task-oriented approach. AMIA Annu Symp Proc 2005:101–5. [PubMed: 16779010]
56. Hersh WR, Pentecost J, Hickam D. A task-oriented approach to information retrieval evaluation. J Am Soc Inf Sci 1996;47(1):50–6.

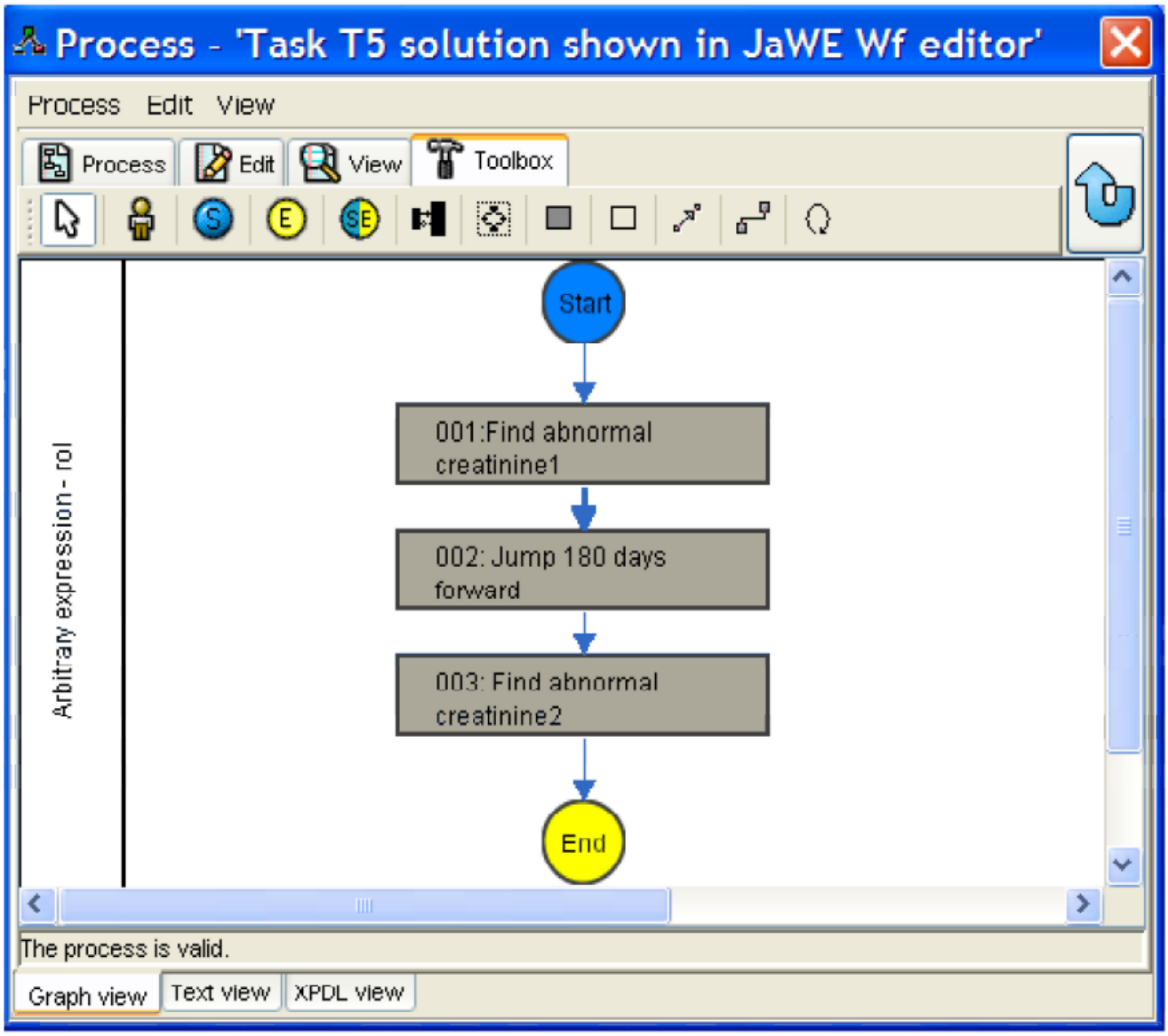

Fig.

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.

Table 1

Evaluation of study subject characteristics

Property	Number (%)
N	18
Sex	
Male	5 (27.8%)
Female	13 (72.2%)
Age	
18-25	1 (5.6%)
26-30	0 (0 %)
31-35	6 (33.3%)
35-40	5 (27.8%)
41-45	1 (5.6%)
45-50	4 (22.2%)
50+	1 (5.6%)
Database experience	
None	0 (0%)
Extremely basic	6 (33.3%)
Basic	4 (22.2%)
Intermediate	8 (44.4%)
Expert	0 (0%)
Source of expertise*	
Medical informatics db class	12 (66.6%)
Db class elsewhere	3 (16.6%)
Self-learning	12 (66.6%)
Educational background*	
Computer science degree	0 (0%)
MD degree	8 (44.4%)
RN degree	5 (27.8%)
MS in medical informatics	4 (22.2%)
PhD in medical informatics	5 (27.8%)
MS in nursing informatics	5 (27.8%)
PhD in nursing informatics	1 (5.6%)
MS in other field	5 (27.8%)
PhD in other field	2 (11.1%)

*Multiple choice is possible for source of expertise background question. The total will thus not equal n.

Table 2

Quantitative evaluation results: mean scores, standard deviations (SD), paired t-test confidence intervals (CI), and *p*-values

	Mean	SD	CI; *p*-value*
Total score (range 0–14)			
SQL total score	6.3	2.2	
RetroGuide total score	11.1	1.9	
Difference RetroGuide-SQL	4.8	1.8	3.4-5.4; $p < 0.001$
Task questions (range 0-9)**			
SQL	4.3	1.6	
RetroGuide	7.3	1.2	
Difference RetroGuide-SQL	3.0	1.3	2.4-3.6; $p < 0.001$
Choice questions (range 0-5)			
SQL	1.9	1.2	
RetroGuide	3.2	1.1	
Difference RetroGuide-SQL	1.3	1.3	0.7-2.0; $p = 0.0004$

* 95% CI, *p*-values are for paired t-test, 2 sided hypothesis.

** Mean scores (from both reviewers) are shown for task questions (T1-T9).

SQL, structured query language.

Table 3

Qualitative evaluation: task questions results analyzed separately for each reviewer

	Mean	SD	CI; *p*-value*
Task questions (reviewer A)			
SQL	4.5	1.6	
RetroGuide	7.8	1.2	
Difference RetroGuide-SQL	3.3	1.4	3.1-3.7; $p < 0.001$
Task questions (reviewer B)			
SQL	4.0	1.6	
RetroGuide	6.7	1.3	
Difference RetroGuide-SQL	2.7	1.4	2.0-3.3; $p < 0.001$

*The range for all items was 0-9 points.

CI, confidence interval; SD, standard deviation; SQL, structured query language.

Table 4

Average ranking of features of a generic analytical tool

Overall rank*	Description
2.22	Modeling paradigm is intuitive even for a non-expert
3.61	Training time required to master the tool is short
5.06	Graphical representation of the problem
5.06	Presence of features which facilitate collaboration of an informaticist/clinician with a professional database analyst
5.22	Query time is short
5.35	Tools incorporate an established technology standard or syntax
5.56	Technology offers direct access data as they are physically stored in the data warehouse
6.06	Technology supports iterative working cycle of a project team (extending previous results and analyses)
8.22	Affordable purchase price for the software
n/a	Other: support for time based structures

* Smaller number indicates higher importance. Only one subject entered an additional free-text feature (exact rank calculation was not possible).

Table 5

Construct reliability, mean and standard deviation (SD) scores (5-point Likert scale)

Construct	Mean	SD	Cronbach's alpha
Performance expectancy (PE)			0.871
PE1	4.45	0.62	
PE2	4.11	0.58	
PE3	3.89	0.76	
Effort expectancy (EE)			0.849
EE1	4.39	0.70	
EE2	4.28	0.75	
EE3	3.89	0.76	
Behavior intention (BI)			0.752
BI1	4.22	0.88	
BI2	4.39	0.70	