A System Supporting Qualitative Research

Emilia Todorova
St Cyril and St Methodius University
of Veliko Turnovo,
Veliko Turnovo, Bulgaria

Dimo Milev
St Cyril and St Methodius University
of Veliko Turnovo,
Veliko Turnovo, Bulgaria

Ivaylo Donchev
St Cyril and St Methodius University
of Veliko Turnovo,
Veliko Turnovo, Bulgaria

***Abstract*—this paper presents a system aimed to be entirely located in the Internet. The database, server side, logic and user interface are accessible regardless the location and equipment of the working team members. The system supports qualitative analysis of large datasets, and the work can be distributed among several team members or teams. A modified mixed method for developing software projects is used, according to the peculiarities of our case, splitting the phase of code writing into two sub-phases each with different number of iterations.**

Keywords—Information systems (IS); Qualitative data analysis (QDA); NoSQL Database; Software Design; Computer Assisted Qualitative Data Analysis Software (CAQDAS)

I. Introduction

The ability to easily gather capacious information on various issues results on one hand results in more accurate and complete data, but on the other needs a relatively easy and convenient way to categorize, evaluate and extract most precise results from the data. Handling massive amounts of data and their processing often dramatically slows down the work. At the same time, attempts to achieve most accurate results require collection of maximum amount of data. Nowadays with the help of information technologies gaining huge amounts of data is an easy task. The hard part is categorization, evaluation and overall extraction of useful information from collected data – in other words, qualitative analysis.

The existence of a software system for organization of qualitative data analysis is of tremendous help to the team1s involved in qualitative research. "Computer assisted data analysis software can be used to make the research process more transparent." [1].

The purpose of this work is to present a cloudy-based system supporting qualitative data analysis. The database, server side, logic and user interface will be accessible regardless to the location and equipment of the working team members.

The system is developed by three groups of graduate students from the Master Course in Information Systems, consulted by lecturers and an expert from an IT company.

II. Defining the Goals

The creation and handling of a system to facilitate qualitative data analysis sets new tasks to developers as well as to trainers in this field. We set three main goals in our work.

The main goal was to develop an on-line system supporting qualitative data analysis, located entirely on the Internet.

The second goal was to explore different methods for developing software projects and choose or propose an appropriate one.

The third goal was to involve in the project Master course students, teachers and IT professionals, and to evaluate the teaching and learning outcomes.

III. Prerequisites

A. Qualitative Research

Some authors emphasize on the research purpose and focus: qualitative researchers are interested in understanding the meaning people have constructed, that is, how people make sense of their world and the experiences they have in the world [2].

Others consider an epistemological point of view: qualitative research is research using methods such as participant observation or case studies which result in a narrative, descriptive account of a setting or practice. [3]

A third group of definitions concentrates on the technology and context of data collection: qualitative research is a situated activity that locates the observer in the world. It consists of a set of interpretive, material practices that makes the world visible. These practices transform the world. They turn the world into a series of representations, including field notes, interviews, conversations, photographs, recordings, and memos to the self. At this level, qualitative research involves an interpretive, naturalistic approach to the world. This means that qualitative researchers study things in their natural settings, attempting to make sense of, or to interpret, phenomena in terms of the meanings people bring to them [4].

Yet there is a simpler and more functional definition suggested by Nkwi, Nyamongo, and Ryan [5]: “Qualitative research involves any research that uses data that do not indicate ordinal values.”

B. Qualitative data

Qualitative data can be arranged into categories that are not numerical. These categories can be physical traits, gender, colors or anything that does not have a number associated to it. Qualitative data is sometimes referred to as categorical data [6].

The project is supported by the Bulgarian Ministry of Education, Project RD-672-08/09.05.2012.

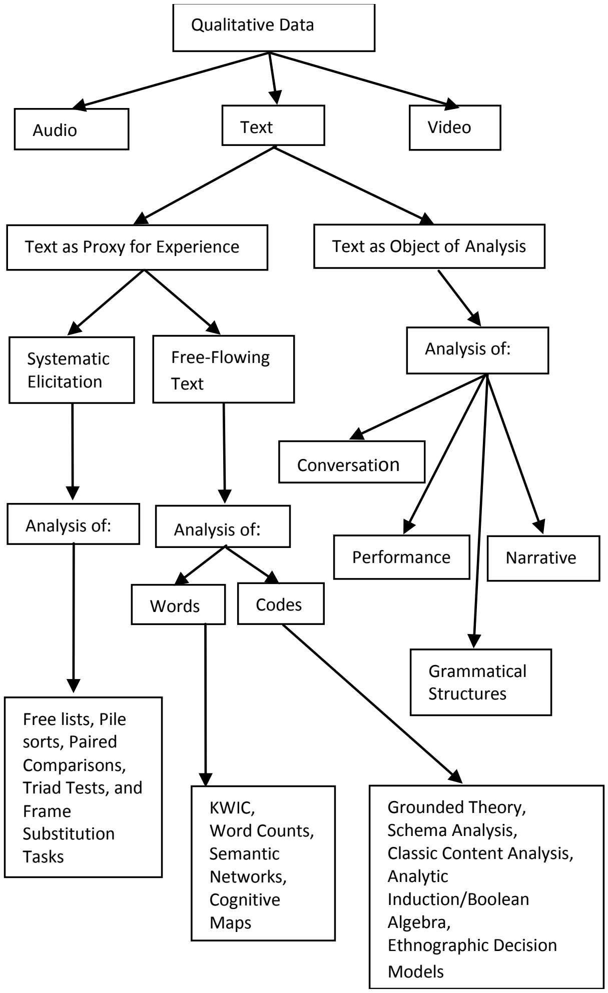
Qualitative Data
Audio
Text
Video
Text as Proxy for Experience
Text as Object of Analysis
Systematic Elicitation
Free-Flowing Text
Analysis of:
Conversation
Analysis of:
Analysis of:
Performance
Narrative
Words
Codes
Grammatical Structures
Free lists, Pile sorts, Paired Comparisons, Triad Tests, and Frame Substitution Tasks
KWIC, Word Counts, Semantic Networks, Cognitive Maps
Grounded Theory, Schema Analysis, Classic Content Analysis, Analytic Induction/Boolean Algebra, Ethnographic Decision Models

Fig. 1. Qualitative data types [7]

Qualitative data is data that approximates and characterizes but basically doesn't measure the attributes of a thing or a phenomenon. It describes data as compared to quantitative data that tends to calculate data. Qualitative has to do with quality and can be subjective. Quantitative has to do with quantity and is measured in numbers.

IV. THE METHOD

The implementation of the software project (information system with integrated database) requires determination of the method of construction. There are two basic styles for developing software projects [8]. One of them is the waterfall method, when the project is partitioned into phases according to the basic activities that are carried out successively. The iterative style sections the project into parts on the base of functional subsets. If we agree that the full software life cycle consists of analysis, design, code writing, testing and evaluation, the first style supposes these phases to be carried out once each, one after another, and the second many times each until the desirable full set of functionalities is reached.

After examining the advantages and disadvantages of existing methods and research experience in this area, we outlined some trends. Most developers especially those involved in object-oriented design do not support the waterfall approach. One of the most important reasons is the difficulty to determine whether the project is developing properly at intermediate stages. It is considered that when using iterative approach the iterations should have fixed time duration (time boxing). When appears that for this period is not possible to finish the planned activities, it is necessary to leave some of the functionalities for the next iteration. This is also a way to prioritize functionalities [8]

For this software project a mixed method is adopted (Fig. 2), The step of writing code is divided into two parts running in parallel: development of the skeleton of the information system and writing modules, implementing specific tools for qualitative data analysis. The special feature here is that the number of embodiments of these two parts may vary prior to the step of testing, such as the expected number of iterations for writing the code of the skeleton of the IS to be less than the number of iterations for the implementation of the functionalities in the qualitative analysis. Naturally it could be expected after a certain iteration skeleton of the information system to be completed, and the inclusion of new tools for qualitative data analysis continues, which means upgrading the functionality of the system.

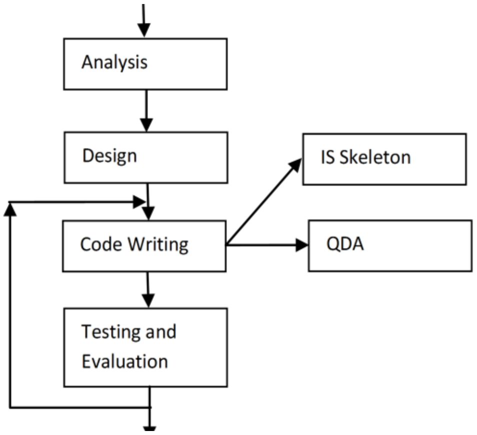

Fig. 2. Method for Developing the Software Project

Determining the iterations of the software project allows separating groups for the development of each phase. There are three groups in the team: the first implements the analysis and design of high level; the second writes code skeleton of the application; the third writes code for the instruments of qualitative data analysis. All groups are involved in the testing and evaluation of the results.

V. IMPLEMENTING THE SYSTEM

A. Defining the Functionalities

The group of analysts examined several CAQDAS packages: Ethnograph, QSRNvivo, Atlas.ti, QDA Miner, Dedoose, DRS (Digital Replay System), NyperRESEARCH, MAXQDA.

The observation and analysis of the methods of qualitative analysis and existing systems helped to define the functionalities of the desirable system.

Functionalities available to users of the system - each user has the following options:

- launch a project for which becomes a manager;
- participate in projects as an analyst, and in this case may assign tags and grades to the paragraphs of the project;
- in case the user has the rights of a project manager he/she can evaluate the work of analysts and edit their grades and labels;

Functionalities provided to Project Manager: each project manager has the following options:

- create a database of paragraphs - one by one or by importing data;
- join or remove members of the team (analysts) to their projects and their assigned rights - analyst, manager;
- create descriptor paragraphs - typical paragraph descriptions that subsequently may be used to group data and analysis results;
- create labels and sub-labels for a project;
- create ratings for the labels;
- delete paragraphs - individually or as a group depending on a specific tag or attribute group (label and grade);
- review the work of each user connected separately and to assess;
- retrieve various reports and graphs based on the processed data;

Features of projects and paragraphs labels:

- database of texts separated into paragraphs;
- tags that analysts assign to paragraphs;
- grades assigned to the labels of a paragraph.

Paragraphs in turn have:

- descriptors - fields describing the paragraph;
- labels and sub-labels;

Labels have grades.

B. The Architecture

The system is built according to the multilayer model architecture. This model is considered to be one of the templates of software architecture. It represents the client-server concept, where the user interface (environment), logic application and work with the database are performed by independent modules. An advantage is that the three-layer model allows to change technology and modify each of the modules without affecting the other layers. The only requirement is to abide by the approved interfaces among the modules. It is admissible the middle (logical) layer to be multi-layer itself, which results in an N-layer model. Fundamental rule in this type of architecture is that the presentation layer (user environment) never communicates directly with the database layer. This makes the model linear.

C. Technologies for Inplementing the Levels

Web technologies are currently the most rapidly developing information technology from the point of view of global connectivity and cloud computing. New technologies in this field appear literally every day and a web programmer needs to stick on the trends. The JavaScript language is well known for Web developers for it is relatively simple but sufficient and powerful. Until recently it was used on1ly for building user interfaces, but lately it finds its place in all layers of programming in multilayer systems. For the implementation of our system we used only JavaScript.

1) The Database

Recently NoSQL databases gain popularity [9]. This type of database provides a mechanism for storage and retrieval of data using free coherent model unlike the more commonly used relational database. The benefits of this approach include simple design, horizontal scaling and subtle control over the information available. Non-relational database is the best optimized repository containing information of type key-value or document with ID. The purpose is to facilitate the processes of recovery, adding information and introduction of excessive amounts of data and to optimize performance in terms of unintentional delays in the system. Since the data that is processed in this project are supposed to be mainly documents, the most suitable model for non-relational database is the type of document repositories. The central concept behind the Document Repository is the notation for "document". Every document-oriented implementation is different according to the details of the definition for "document", but in all implementations the documents encapsulate and encode data (or information) in any standard formats. The used formats are: XML, YAML, JSON and binary formats like BSON, PDF and Microsoft Office documents (MS Word, Excel, etc.).

Different implementations offer different approach to organizing and grouping of documents:

- Collections;
- Tags;
- Invisible metadata;
- Hierarchy of directories;

The address of a document is represented in the database by a unique key that identifies the document. One of the other characteristic features of document-oriented databases is that besides the simple search by key-document or key-value, which can be used to retrieve the document, databases provide user interface and query language that allow documents to be opened according to their content. In view of our goal to use only JavaScript, we have chosen MongoDB for the development of our system. This database stores the documents in JSON format, which is natural for JavaScript and allows displaying the data in an appropriate form. It is easily readable by developers and is easily parsed by various programming languages. JSON is often used for serialization and transmission of structured information and this implementation is much shorter in code than XML. MongoDB uses JavaScript for querying and administration as well. This makes it particularly suitable for our case.

2) *The Serverside and Logic*

As for now perhaps the only technology that allows use of JavaScript for writing server-side applications is Node.js. Node.js platform is developed on the Chrome JavaScript environment in order to allow building fast and easily scalable network applications. It uses an event-driven model, which does not block the input and output of the system. This makes it light and efficient and is a suitable choice for applications with very intense real-time data processing and multiple devices handling simultaneously. Another advantage is the large set of additional packages that facilitate programmers work and perform many different tasks: web servers, access to file systems, access to ports, database drivers and much more [10], [11].

Characteristic feature of the event-driven model of Node.js in asynchronous non-blocking mode is that it runs in only one thread, but it still executes parallel and simultaneous operations for requests from many users.

The model that we use for interaction between client and server-side is ReST (Representational State Transfer), a distribution system framework based on Web protocols and technologies [12]. ReST architectural model includes interactions between server and client during the data transfer. The most commonly used module for building ReST architecture is Node.js Express. This is a ready module for building a web server with complete infrastructure for organizing views.

3) *User Interface*

JavaScript is widely used for building user web interface. There are too many software frameworks that generate HTML document templates. On the first iteration phase we chose the simplest template – Mustache. For our purposes we use mu2Express, as we use Express upgrade to organize our web server and its ReST functionality. This is a simple template for image description. Keywords between braces are replaced by data derived from our database.

VI. Future Work

Testing and evaluation of the system is forthcoming. The development of the modules implementing instruments for QDA will involve longer period of multiple of iterations. And yet the teaching and learning outcomes of the experiment have to be evaluated.

VII. Conclusion

In light of the current requirements for achieving independence from platforms and geographic location of the members of research teams a method for creating cloud-based information system is suggested. A model of the system is developed and technology for its implementation is proposed. The proposed technology is universal and platform-independent and provides easy upgrade and maintenance. The suggested method for developing the project facilitates the continuous development of the system.

References

[1] R. Mary. Making visible the coding process: using qualitative data software in a post-structural study. Issues in Educational Research. Vol.19 (2), 2009, p.142.

[2] S. Merriam. Qualitative research: A guide to design and implementation. San Francisco, CA: Jossey-Bass. 2009.

[3] G. Parkinson, R. Drislane. 2011, Qualitative research. In Online dictionary of the social sciences. http://bitbucket.icaap.org/dict.pl

[4] N. Denzin, Y. Lincoln. (Eds.) Handbook of qualitative research (4th Ed.). Thousand Oaks, CA: Sage. 2011.

[5] P. Nkwi,I. Nyamongo, G. Ryan. Field research into socio-cultural issues: Methodological guidelines. Yaounde, Cameroon, Africa: International Center for Applied Social Sciences, Research, and Training/UNFPA. 2011.

[6] C. Taylor Qualitative Data: at http://statistics.about.com/od/Glossary/g/Qualitative-Data.htm/

[7] Ryan, G., & Bernard, R., 2000, Data management and analysis methods. In N. Denzin & Y. Lincoln (Eds.), *Handbook of Qualitative Research* (pp. 769–802). Thousand Oaks, CA: Sage.

[8] M. Fowler. UML distilled. Third edition. Addison Wesley, 2004.

[9] P.J. Sadalage, M. Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley, 2012.

[10] M. Kiessling. The Node Beginner Book. Leanpub, 2011.

[11] Node.js v0.10.26 Manual & Documentation. http://Node.js.org/api/

[12] R.T. Fielding. Architectural Styles and the Design of Network-based Software Architectures. Dissertation, Chapter 5. University of California, Irvine. 2000.