UNIVERSITY OF GOTHENBURG

Executing statistical programs from the Database management system

KHATSANOVSKYY VADYM

Bachelor of Science in Software Engineering and Management Thesis

Report No. 2010:083
ISSN: 1651-4769

University of Gothenburg
Department of Applied Information Technology
Gothenburg, Sweden, May 2010

Abstract

With the evolution of information technology, the process of gathering and storing data reached an unprecedented level. It resulted in an increasing need for effective and efficient systems that are able to handle massive amounts of information. Hospital registers, national censuses and other surveys constantly produce outstanding volumes of microdata that has to be stored and analyzed. In this thesis, we will analyze systems that are designed for statistical analysis of microdata. We present a novel approach to the architecture of such systems and a prototype that implements it.

Keywords: Database management system, Statistical package, microdata, protected remote statistical analysis.

1 Introduction

Statistical analysis of large data sets forms a foundation for multiple research projects in various scientific fields. It represents analysis of numerical data related to individuals or experiments and uses microdata, as the main source. The common feature of all microdata is that it includes identifying information for specific individuals making it possible to track potentially sensitive information back to real people.

There are many different approaches that are aimed to satisfy need of controlling access to microdata, while still allowing access to these large and important datasets to as many researchers as possible thus enhancing our ability to find solutions to some of societys greatest problems. Confidentiality rule should be followed during the process of statistical analysis. Executing remote statistical analysis is one of the ways to prevent the unnecessary disclosure of microdata and guarantee confidentiality.

Nowadays, there exists a number of systems, e.g. LISSY[1] , BioGrid Australia[2], MONA[3], which give a possibility to conduct statistical analysis on a remote location of the data owner. All of the above mentioned systems are used by national statistical agencies and involve high development and operational costs. The aim of our research is to decrease these costs and allow smaller data owners to provide access to valuable research data. Ans as a result to increase the amount of microdata available for researchers. We try to access our aim through design and implementation of the novel architectural approach.

Existing systems for remote statistical analysis share common architectural pattern and work concept. The overall control over statistical analysis is assured by two separate subsystems: data management system, used to control access to data; additional system that controls users access to statistical packages and execution of statistical analysis. On the contrary, we propose to use DBMS for control over access to data, as well as for control over access to statistical packages and execution of statistical analysis. In this way we eliminate the need to purchase or develop any additional systems. Our solution is designed to preserve the main characteristic of the remote statistical analysis systems. That is to return only results of the statistical analysis to the user, while microdata remains at the location of data owner.

We use design science research method to implement a prototype named SAQeL[4]. It is aimed to extend DBMS by integrating it with statistical package in order to provide facilities for statistical analysis. The scope of our research is defined by the usage of RDBMS IBM

[1] `http://www.lisproject.org`
[2] `http://www.biogrid.org.au`
[3] `http://www.scb.se`
[4] Statistical Analysis from SQL

DB2[5] and statistical package SAS[6]. The results of the study show that our architectural approach can be used in real environment and that the SAQeL prototype can be used as a foundation for further development of industrial scope systems.

The remainder of this paper is organized in the following way. Section 2 describes theoretical background and related work. Section 3 defines the research methods used. Section 4 gives a proposal for architecture and the rationale for its use. Section 5 outlines system prototype. Section 6 presents concluding remarks.

2 Background and related work

In this section, we introduce theoretical background to the problem area of our research and give an overview of the related studies.

2.1 Theoretical background

Statistical analysis is aimed to confirm or falsify specific hypotheses. It includes two main tasks: data preparation and analysis itself. The conclusions of the statistical analysis are based on the outputs received in form of scalar values, tables or graphs.

Statistical packages provide environment for statistical analysis. The core of statistical package is a library of software functions that implements a variety of statistical algorithms. Usually, each package has internal problem-oriented programming language used to write *statistical programs* defining what kind of statistical operations should be performed on data and in which sequence. To facilitate the process of statistical programs development the majority of currently used statistical packages provide IDE. There are general statistical packages that support a variety of statistical procedures, among them SAS, R[7], STATA[8]. At the same time, some packages are aimed for specific purposes, e.g. time series analysis, probability distributions etc. For example SPSS package[9] , performs statistical analysis specific for sociological research.

The process of statistical analysis can differ depending on type of microdata involved. When there are no restrictions on access to the microdata, the process of statistical analysis starts with the execution of queries aimed to prepare the desired microdata subset. Queries can be executed in a two ways: directly from DBMS using standard SQL language; or from the statistical package using data management language specific to it, e.g. SAS SQL [SAS software, 2009]. When the process of data preparation is over, it is time to start analysis. Researcher writes and executes statistical program using specific internal language of the statistical package. During execution, statistical programs generates the results of statistical analysis in form of scalar values, tables or graphs.

When statistical analysis is performed over sensitive microdata, it is important to keep such data at the same location and exclude its transfer to the researcher [10]. The systems for *remote statistical analysis* provide facility, where researcher can submit queries for statistical analysis from their own computer. The queries are handled on a remote location of the microdata owner and generated results of the statistical analysis are returned back to the researcher [Sparks et. al., 2008].

[5] `http://www.ibm.com/software/data/db2/`

[6] `http://www.sas.com`

[7] `http://www.r-project.org/`

[8] `http://www.stata.com`

[9] `http://www.spss.com/statistics/`

[10] Further in the paper, person conducting statistical analysis is interchangably referred to as "user"/"researcher"

2.2 Existing systems for remote statistical analysis

There are a number of systems that are designed to execute remote statistical analysis of microdata. Their main goal is to exclude unauthorized access and to secure confidentiality. Although, they may differ according to the area of application, two basic concepts of remote statistical analysis are used [Fellegi et. al, 2007]:

- **remote execution:** researcher submits statistical program and receives the output later over the Internet. Such systems have special submission facilities and the researcher doesn't get direct access to the remote analysis server. Statistical analysis is executed in batch mode. One of the examples is LISSY from Luxembourg Income Study;
- **remote facilities:** researcher performs the analysis and has immediate access to the answer on the screen. In this case, researcher gets direct access to the remote analysis server and works with statistical software interactively. Examples of such systems are BioGrid Australia at Melbourne Health, MONA at Statistics Sweden, the Danish system at Statistics Denmark.

Another important characteristics of the above mentioned systems are methods for data access. Most of them are designed to extract data from the original database and store it in files, before the analysis is performed. Only BioGrid Australia allows statistical programs direct access to the DBMS for data extraction.

Some of this systems provide additional control to reveal any attempts for extracting sensitive data during analysis. It can be done both before the statistical analysis is performed or afterwards. The Danish system provides control over the analysis results to spot possible disclosure, while LISSY also tests submitted statistical programs in search for "illegal commands" [11].

LISSY was designed to analyze economic data in support of Luxembourg Income Study. It is one of the first system that was designed to provide facilities for remote statistical analysis. This system can be used by registered users, who have two options in making request for statistical analysis: sending statistical program formatted according to the given pattern by e-mail; usage of web-based Job submission interface. The system identifies user and tests program for the use of illegal statistical commands. If illegal commands are identified, the user receives an error message, explaining the violation. If the results of statistical analysis are considered suspicious, they are sent to the system administrator for manual check-up. Otherwise, the results of the statistical analysis are sent back to the user by e-mail irrespective of the submission alternative used [Barry & Marc, 2003].

BioGrid Australia was designed for analysis of health data from several Melbourne hospitals, Australia. This system uses SAS Enterprise Guide [12] as the user interface. SAS also provides authorization and authentication. System provides possibilities to integrate data from different sources using federated databases [Hibbert et al., 2007].

MONA was designed to analyze data from Statistics Sweden. The system provides secure Internet access to a remote desktop[13] from a Windows or Unix client. According to the user

[11] Aimed to add sensitive data to the results of analysis

[12] `http://support.sas.com/documentation/onlinedoc/guide/`

[13] `http://en.wikipedia.org/wiki/Remote_Desktop_Protocol`

preferences, the desktop may contain different statistical packages to choose from: SAS, SPSS, STATA, GAUSS[14] etc. So the user can work with the packages as if from the local computer. All computations are done remotely and each user is provided with the space for file storage. This enables data to stay on site at Statistics Sweden. Results of the statistical analysis can be moved to a special folder and consequently sent to the user's e-mail [Hjelm, 2005].

The Danish system was designed to analyze data from Statistics Denmark. It enables analysis with the following statistical packages SAS, SPSS, STATA, GAUSS, etc. The user can access the Unix environment of Statistics Denmark from his/her own workplace. Communication is encrypted by means of a RSA SecurID [15]card. The results of analysis are stored in a special file, which is later transferred to the user via e-mail. The e-mails are checked manually by the Research Unit Service and the user is contacted in case the requested data was too detailed [Borchsenius, 2005].

2.3 Existing systems that extend DBMS functionality

Another research area, which we consider to be relevant, is represented by projects that extend DBMS, by implementing interface to call analysis programs, e.g. La Select, or external statistical functions, e.g. MECHAMOS, from it.

La Select was designed to process earth-science data. It uses SQL-like language that enables execution of image analysis programs. The user can issue queries to access distributed data and perform execution of various image analysis algorithms on such data. For example, the data can be represented by satellite images and analyzed according to the image manipulation algorithms. The results of the analysis are presented by single table and stored in DBMS [Luc et al., 2001].

MECHAMOS was designed to provide multibody analysis and is based on the object-relational data management system AMOS II[16]. To provide additional mathematical functionality, AMOS II has been extended with client-server connection to Matlab and MapleV. At the same time object-oriented query language AMOSQL was extended to give the user possibility of issuing queries that perform multibody system analysis. Due to the limitations of Matlab[17] and MapleV[18], some of the results can not be stored back to the database and are instead stored in a file system [Tisell & Orsborn, 2000].

3 Research method

This section formulates the research problem and provides information about research environment and limitations. It also gives an overview of the research methodology and data collection.

3.1 Research problem

Trying to find a solution for the problems discussed in introduction, research question could be formulated as *"How to extend functionality of DBMS so that it can be used as a system for statistical analysis of microdata?"*. Possible solution is aimed to make a bigger amount of databases available to researchers, preserving the necessary condition of keeping confidential data about individual people. We try to figure out if it is a feasible task; if security

[14] `http://www.aptech.com/gauss.html`
[15] `http://www.rsa.com`
[16] `http://user.it.uu.se/~udbl/amos/`
[17] `http://www.mathworks.com/products/matlab/`
[18] `http://www.maplesoft.com/`

and data protection requirements can be fully satisfied by built-in DBMS facilities; what are the benefits and drawbacks of such solution, comparing with other systems.

At the same time, this purely technical question should be addressed keeping in mind other important problems. One of them - emphasis on usability, because the problem is tightly connected with working process and there is always a trade off between usability and security. The solution will influence the way how people work, so it is necessary to improve the security of the analysis process with the minimum damage on its usability.

3.2 Research environment

The research took place at Department of Medical Epidemiology and Biostatistics at Karolinska Institute in Stockholm. This Thesis work is related to a large project CODIR (Cross-Organizational Database Infrastructure for register-based Research), which is held by the same department. CODIR is aimed to develop efficient, scalable and secured infrastructure for scientists to perform research on sensitive data from registers and other sources stored in different authorities and organizations [Fomkin et al. 2009]. Such infrastructure requires inner statistical analysis to be provided.

3.3 Research limitations

The scope of the research was limited by the choice of specific statistical package and DBMS. We used the general statistical package SAS. It is considered to be industrial standard for statistical analysis, being used at more than 45,000 sites in over 100 countries including 92 of the top 100 companies of the 2009 [SAS Software, 2009]. Also, it is widely used by researchers of Karolinska Institute and therefore we worked with SAS Base [19] v. 9.2. The choice of DBMS was made in favor of RDBMS IBM DB2 v. 9.5.

3.4 Research methodology

As the research question of this thesis is problem-centered and real-world oriented, it can be related to the pragmatic school of philosophy. "Pragmatism adopts an engineering approach to research it values practical knowledge over abstract knowledge, and uses whatever methods are appropriate to obtain it"[Hevner et al., 2004]. Such approach gives to the researcher total freedom in the choice of methods and their combination, as the main goal remains finding the most "truthful" solution.

That is why methods with pragmatism as underlying philosophy were under primary consideration. Because the aim of our research is to create and evaluate the system prototype, we consider design science research most fitting. The main difference between routine design and design research is that the last one addresses unsolved problems in innovative way or solved problems in more effective way [Hevner et al., 2004]. And it is true for our research, as our main idea is in exercising novel architecture as an alternative to existing approaches.

The outcomes of the design science research are never represented by ready-to-use systems, but form a starting point for later implementation of the industrial scale systems [Denning, 1997]. Design science research is usually performed inside target organization, but its outcomes can be efficiently used by other organizations as well.

Hevner [Hevner et al., 2004] presents concep-

[19] `http://www.sas.com/technologies/bi/appdev/base/`

tual framework and 7 guidelines "for constructing and evaluating good design science research". We use them for further presentation of our research.

1. Design as an artifact. Artifacts constructed in design science research provide "proof by construction" [Nunamaker, 1997]. They demonstrate the feasibility of design process and product. As it is common for design science research, our system prototype is not suited for direct application. It was developed to help evaluate the possibility of applying novel architectural approach to the problem of executing remote statistical analysis.

2. Problem relevance. "Business problems and opportunities often relate to increasing revenue or decreasing cost trough the design of effective business process". [Hevner et al., 2004] Our research is aimed to decrease the development and operational costs of remote statistical analysis systems and therefore to make such systems accessible for smaller institutes and data holders.

3. Design evaluation. The utility of the artifact can be proved via successful execution of evaluation methods. There are several evaluation methods that are characteristic for design science research. We consider observational and descriptive to be appropriate for our research. Observational evaluation method implies studying the functioning of the artifact in real environment. Accordingly, we used the real database and statistical programs that are applied by researchers of Karolinska Institute to test our prototype. Descriptive evaluation method use related research to built arguments for the artifact utility. And the detailed studies of the relevant systems gave us the possibility to compare our approach to the solutions constructed for the same problem.

4. Research contributions. The main contribution of the design science approach is the artifact itself. It can be aimed to extend the knowledge base or apply existing knowledge in new ways. Prototype demonstrates feasibility of extending DBMS, so it can be used for executing statistical programs. The detailed research of related studies showed that it was never done previously, so our prototype in itself is a contribution to the design science.

5. Research rigor. Rigor of the research measures how strong is the strength of theoretical foundations and research methodologies used to construct the artifact. Prior research in developing systems for remote statistical analysis serves as a foundation of our research and deficiencies of these approaches form our motivation.

6. Design as a search process. Our research has strictly defined time scope that cannot be prolonged under any circumstances. The iterative nature of the design science research makes is suitable for development of the prototype in the short period of time and in the scarce previous domain-specific knowledge. It gives a possibility to use trial-and-error search and this characteristic was utilized in our case.

7. Communication of Research. "Design-science research must be presented both to technology-oriented as well as management-oriented audience" [Hevner et al., 2004]. A presentation of the results of our studies was organized at Karolinska Institute. It involved both technical workers who deal with databases, researchers who are hypothetical users and managers who are in charge to make decision about the relevance of this research and its practical application.

3.5 Data collection

This research included following data collection activities:

- studying of the system documentation and APIs: As we discussed earlier, IBM DB2 is used as DBMS and SAS Base as statistical software. Naturally, development of the system prototype, which combines those systems, requires deep knowledge of system documentation and also collecting data about APIs.

- search for the existing knowledge within the problem scope: Helps to incorporate organizational culture into solution. There is always a risk of producing a secure system which no-one will use. That is why, being inside the organization is essential. Close communication can reveal how people work and how to make qualitative changes less painfully.

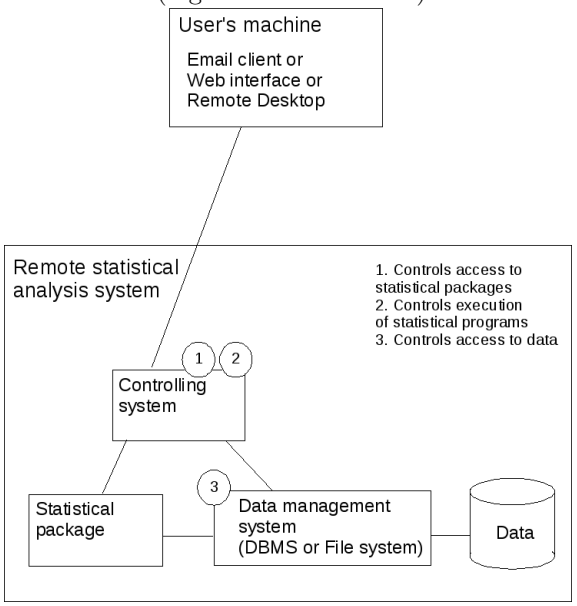
4 Architecture

In this section, we analyze typical architecture implemented previously in the currently used systems, see section 2.2, and describe novel architectural proposal developed as the solution for the research problem stated in the section 3.1.

4.1 Previously implemented architecture

The architecture, which we describe in this section, is common for the systems described in section 2.2 According to Fig.1, the researcher uses E-mail client/Web interface/Remote desktop to interact with the remote statistical analysis system. All interaction with user is handled by a specific *Controlling system*. (1) It performs user authentication and authorization as well as controls execution of statistical analysis. (2) While execution of statistical programs, *Statistical package* accesses data from the *File system/DBMS* . (3) *File system/DBMS* performs authorization of access to the data. Therefore, overall control of the process of statistical analysis is spread over several subsystems.

Figure 1. Previously implemented architecture (high level abstraction)



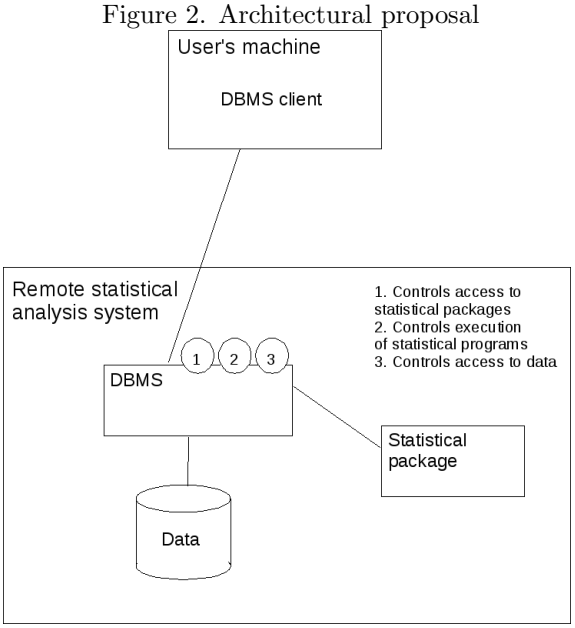
4.2 Architectural proposal

The main difference of our architecture is that we propose to use DBMS to call Statistical Package. DBMS manages all three activities described in the previous section. Build-in authentication, authorization and auditing mechanisms of DBMS are used to control both access to data and access to statistical packages. Functionality of DBMS is extended to call statistical programs from it.

Our architectural proposal is presented in Fig. 2. Researcher uses a standard *DBMS client* to issue queries, create database views and execute statistical analysis. The first step is submission of the SQL query for creating database views to the *DBMS* with the help of *DBMS client*. This query will only be executed

if the process of authorization finished successfully. After that, the researcher creates statistical program that is registered in *DBMS* and requests its execution. In case researcher is authorized to perform statistical analysis, *DBMS* submits the program to the *Statistical package* for execution and transfers all the data necessary for execution. After the execution of statistical program, results of the statistical analysis are transferred back to the researcher.

Figure 2. Architectural proposal



5 Prototype

In this section, we investigate proposed architecture by implementing a *SAQeL* prototype Statistical Analysis from SQL. We present design considerations, architecture, and an example of the running statistical analysis.

5.1 Design Considerations

SAQeL design is based on simplicity. This made it possible to try several approaches to an existing problem in a short period of time and choose the most suitable solution. At the same time, it made our prototype scalable and enabled development of industrial system on its base. The following decisions influenced the architecture of our prototype.

How to call statistical analysis in SAS from external system?
There are two ways of calling SAS package from other systems:

- to use "SAS/integration technologies"[20];
- to execute it in a batch mode.

"SAS/integration technologies "provide large collections of APIs that enable integration using external applications. However, binding our solution to it would require later alterations in case of the need to work with other statistical packages, e.g. STATA, R etc. To make our solution universal for all kinds of statistical packages, the choice fell on a batch processing using SAS Base.

At the same time, to secure more control over the execution process of SAS Base, we decided to include additional logic to every SAS program. For example, code responsible for redirection of the results to *SAS output delivery system* [21] is added to each SAS file.

How to transfer data from DB2 to SAS?
One of the most important decisions we had to make was how the statistical program should access datastored in DBMS. Two approaches were considered:

- to export data from the database into a file and read that file later from a statistical program;
- to access data directly from a statistical program via database connection.

[20] `http://www.sas.com/technologies/bi/appdev/inttech/`

[21] `http://support.sas.com/rnd/base/ods/index.html`

The first approach can increase time required for analysis, which is especially noticeable in case of big volumes of the data. It also has negative impact on data security. The second approach requires that statistical package provides possibility to access database from the statistical program. To our knowledge, at least such packages as SAS, R and STATA support second approach, so we decided to adopt it.

How to call SAS Base in a batch mode from the DB2?

There is a standard way to extend functionality of the DBMS by developing and deploying *routines*. At the same time, our work was significantly influenced by two restrictions that refer to all DB2 routines:

- it is not possible to create new threads or processes from a DB2 routine;
- it is not possible to create connection to the DBMS from a DB2 routine.

According to the reasoning in previous two questions, the decision was made to execute statistical program in a batch mode and to access data from it via additional connection. But the above-mentioned restrictions make it impossible to trigger execution of such statistical program directly from a DB2 routine. Therefore, there is a need to introduce additional component between DB2 and SAS Base. As such component, we decided to contract a standalone server-like process, which interacts with DB2 routine via socket connection.

Which type of DB2 routine to use?

We considered the following routines, supported by DB2: *user defined functions* and *stored procedures*. They possess several characteristics important for our research:

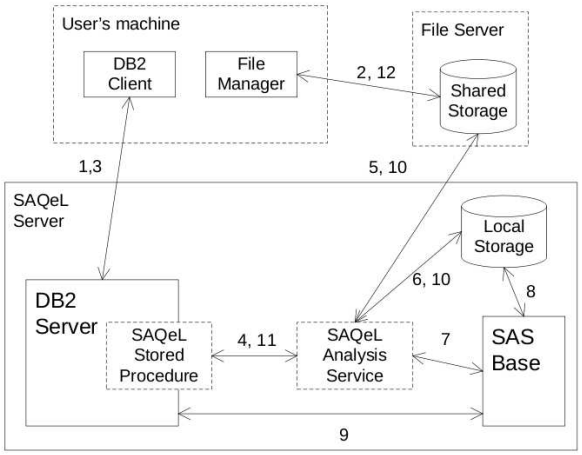
- The restrictions on parameters taken by the routine: their quantity and type;
- In which form the results of routine execution are returned: rows, tables or scalar values;
- Constraints for logic, performed by routine, e.g. its ability to use external libraries.

All of these characteristics vary depending not only on the type of routine, but also according to the programming language used to implement it. So, we had to analyze what restrictions the combination of programming language with above mentioned types of routine can produce. We found that *Java stored procedures* are the most suitable, considering above-mentioned required characteristics.

5.2 SAQeL Architecture

SAQeL architecture is presented in Fig.3. Both *DB2 Server* and *SAS Base* run on the same physical server, *SAQeL Server*. *DB2 Server* is extended with an external stored procedure, *SAQeL Stored Procedure*, implemented in Java. *SAQeL Stored Procedure* is deployed on *DB2 Server* and can be invoked using SQL commands.

Figure 3. SAQeL prototype



The primary goal of *SAQeL Stored Procedure* is transmitting parameters of a statistical analysis execution from a user to *SAQeL Analysis Service* and returning the status of the statistical analysis execution back to the user. *SAQeL Analysis Service* is a Java process that runs permanently. It executes SAS program in a batch mode, i.e. without users direct interaction with *SAS Base*. For this purpose, *SAQeL Analysis Service* prepares configuration parameters for batch execution and manages the statistical analysis results. *SAQeL Analysis Service* receives statistical analysis requests from the *SAQeL Stored Procedure* via a TCP/IP connection socket.

User communicate with *DB2 Client* and *File Manager* to submit their analysis for execution on *SAQeL Server*. *File Manager* is used to store SAS programs on storage shared between Users Machine and *SAQeL Server*, and to access the analysis results from there. *DB2 Client* is used to issue SQL queries, which perform data analysis, and to access information about successful execution of the analysis or occurrence of an error.

The process of an analysis is as follows. Through *DB2 Client* a user (1) issues query to *DB2 server* for creating views to be used in SAS programs. Then a SAS program is created by the user and (2) saved at *File Server*. The user makes request (3) to perform statistical analysis by calling *SAQeL Stored Procedure*. As input parameters of this procedure, the user specifies the name and location of the SAS program, credentials and desired location for storage of the analysis results. *SAQeL Stored Procedure* (4) transforms input parameters into a message, initiates a socket connection with *SAQeL Analysis Service* and transfers this message. *SAQeL Analysis Service* determines the authenticity of the request, by checking its compliance with the internal protocol. In the case of positive result, it extracts information from the received message, (5) reads the user's original SAS program from *File Server* and generates a SAS program for execution by adding supplementary configuration code to the original SAS program. The generated SAS program (6) is saved in a temporary file at *Local Storage*. After that, *SAQeL Analysis Service* (7) makes system call to the operating system to run *SAS Base* in a batch mode. *SAS Base* (8) reads the file with the SAS program from *Local Storage* and executes it. During the execution, it (9) creates a connection to *DB2 Server* and accesses the data. When the statistical analysis is completed, control is returned back to *SAQeL Analysis Service*. *SAQeL Analysis Service* (10) moves the analysis results from *Local Storage* to *File Server* if the execution was successful. Then *SAQeL Analysis Service* generates output message with the general information about the execution results or an error, and (11) sends the message to *SAQeL Stored Procedure*. *SAQeL Stored Procedure* presents the message about completion to the user in *DB2 client*. Finally, the user (12) accesses the analysis results using *File Manager*.

5.3 An Example of Running Analysis

As mentioned previously, we have used real data to test SAQeL prototype. The following is an example of doing survival analysis using SAQeL prototype for a study from epidemiological research on cervical cancer.

First a researcher creates views, which are going to be accessed in a SAS program. For example, view survmig.pc_cohort is created by:

```
CREATE VIEW survmig.pc_cohort AS
SELECT lopnr, diagyr FROM
(SELECT lopnr, MIN(diag_cancer_yr) AS
diagyr FROM cerv_db.cancer
```

WHERE icd_7='171'
AND malign_benign IS NULL
GROUP BY lopnr)
WHERE diagyr BETWEEN 1960 AND 2005

Then the user writes a SAS programs in terms of a view survmig.pc_cohort_duration as following:

PROC LIFETEST DATA=
survmig.pc_cohort_duration METHOD=km
PLOTS=(s) NOCENS;
*TIME years*censor(1);*
STRATA birth_place;
RUN;

The SAS program is stored in *Shared Storage*.

Finally, the user calls *SAQeL Stored Procedure* with following parameters: user name and password for DB2 connection form SAS, name of the SAS program, and input and output paths:

CALL SAQeL (john, abc123, mysasprogram,
S:\john\analysis\Programs\,
S:\john\analysis\Results\);

After the analysis execution the user receives a message in DB2 client, which notifies that the analysis has been completed and the results are available in the specified folder.

6 Conclusions

All of the existing systems for the remote statistical analysis of the microdata involve high costs and, as a result, these data security solutions are used only by large governmental organizations. We have investigated the common architecture of such systems and proposed a novel architecture for executing statistical analysis. Based on proposed architecture, we have implemented SAQeL prototype that demonstrates interface between DB2 and SAS.

According to Barry and Marc [2003] the main quality attributes that should be considered while implementing system for remote statistical analysis is confidentiality, user friendliness and feasibility of implementation. Our approach is more feasible than the existing due to the lower costs for implementation and maintenance, without compromising confidentiality and usability. Utilization of the DBMS privacy protection is the key to substantial decrease of the implementation time and required maintenance and, therefore, costs. Our experience has proved that such system is much simpler to implement. So, simplicity is the key to adaptability and extensibility in other systems like this.

This new approach enables development of the remote statistical analysis systems at reasonable costs and, therefore, makes them accessible for smaller organizations, giving them possibility to provide access to their data sets. From the social perspective, this solution will allow us to broadly extend access to large scientific data sets without compromising the privacy of people directly involved. The scientific impacts are significant, as access to additional information gives researchers greater opportunity to discover new solutions.

We can outline the following steps that could be taken for further development of the alternative remote statistical analysis system based on our research:

Adding statistical packages. Currently SAQeL prototype gives a possibility to conduct statistical analysis using only SAS software, unlike other systems that execute statistical programs using several statistical packages. We consider that such extension can be done within the same approach as used for SAS, by executing statistical programs in a

batch mode and creating additional connection to the database from the inside of the statistical program.

Transferring results via database channel. One of the main disadvantages of the SAQeL prototype is that it does not transfer the results of the analysis to the user's computer. Instead, they are saved in a remote file system. Additional research is required to determine in which way results can be transferred to the user. Our proposal is to use the same database channel, which was used to issue the query.

Authentication improvement. Currently, the user is supposed to specify user name and password in the query. SAS uses them later to access the database. Though such procedure has its disadvantages, being inconvenient for the user and decreasing security. To eliminate them, it is necessary to develop alternative authentication procedure that will not require input of credentials in each query. One of the possible solutions is to run statistical packages as a trusted DBMS process.

Control over the returned results. Our solution provides no additional control over the content of returned results, unlike existing systems that provide control over the availability of sensitive data in the returned results. We consider that SAQeL prototype does not allow extending for manual check up of the results of statistical analysis. However, we see the implementation of the automatic check up as a good alternative.

Acknowledgements: I thank my supervisors, in particular, Ruslan Fomkin at Karolinska Institute and William Eugene Sullivan at IT-Univeristy for all the help in conducting research and improvement of this paper.

I am grateful to professor Jan-Eric Litton and all of the MEB staff for giving me this grate experience.

References

[Barry & Marc, 2003] Barry, S., Marc, C.: *Remote access systems for statistical analysis of microdata.* Statistics and Computing 13 (2003) 381-389.

[Borchsenius, 2005] Borchsenius, L.: *New developments in the Danish system for access to micro data.* Monographs of official statistics (2005) 13-20

[Denning, 1997] Denning, P. J., *A New Social Contract for research,* Communications of the ACM 40:2, pp. 132-134, 1997

[Fellegi et. al, 2007] Fellegi, I. et al., *Managing Statistical Confidentiality & Microdata Access, Principles and guidelines of good practice.* Conference of European Statisticians, 2007.

[Fomkin et al. 2009] *Federated Databases as a Basis for Infrastructure Supporting Epidemiological Research*, Ruslan Fomkin, Magnus Stenbeck, Jan-Eric Litton, 20th International Workshop on Database and Expert Systems Application

[Hevner et al., 2004] Hevner A.R., et. al., *Design science in information system research*, MIS Quartely Vol.28 No.1 pp. 75-105, 2004

[Hibbert et al., 2007] Hibbert, M., Gibbs, P., O'Brien, T., Colman, P., Merriel, R., Rafael, N., Georgeff, M.: *The Molecular Medicine Informatics Model (MMIM).* Stud Health Technol Inform 126 (2007) 77-86.

[Hjelm, 2005] Hjelm, C.G.: *MONA-Microdata ON-Line access at Statistics Sweden.* Monographs of official statistics (2005) 21-28.

[Luc et al., 2001] Luc, B., Fran, oise, F., Fabio, P., Patrick, V.: *Processing Queries with Expensive Functions and Large Objects*

in Distributed Mediator Systems. Proceedings of the 17th International Conference on Data Engineering. IEEE Computer Society (2001) 91-98

[Nunamaker, 1997] Nunamaker, J., et. al., *Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings*, Journal of Management Information Systems, (13:3), Winter 1996-97, pp. 163-207.

[SAS Software, 2009] SAS Software, *SAS company overview* `http://www.sas.com/corporate/overview/index.html`

[SAS software, 2009] SAS Software, *SAS(R) 9.2 SQL Procedure User's Guide* `http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/a001407955.htm`

[Sparks et. al., 2008] Sparks, R., Carter, C., Donnelly, J.B., O'Keefe, C.M., Duncan, J., Keighley, T., McAullay, D.: *Remote access methods for exploratory data analysis and statistical modelling: PrivacyPreserving Analytics.* Comput Methods Programs Biomed 91 (2008) 208222

[Tisell & Orsborn, 2000] Tisell, C., Orsborn, K.: *A system for multibody analysis based on object-relational database technology.* Advances in Engineering Software 31 (2000) 971-984