RUSSIAN ACADEMY OF SCIENCES
PETERSBURG NUCLEAR PHYSICS INSTITUTE

Data acquisition and control software of the STAR and PHENIX gas systems

P. Kravtsov

Gatchina 2005

УДК 681.513.1-2, 533.27

Программное обеспечение для управления газовыми системами экспериментов STAR и PHENIX

П.А. Кравцов

Аннотация

Рассматривается комплекс программного обеспечения для управления газовыми системами детекторов TEC/TRD, DC/PC, MuID и MuTr в эксперименте PHENIX, детектора TPC в эксперименте STAR и тестовой газовой системы детектора ATLAS CSC. Несмотря на отличия газовых систем, применение объектной многозадачной модели позволило создать программный комплекс, ориентированный на абстрактную газовую систему, который легко адаптируется к любой из вышеперечисленных систем.

Abstract

The DAQ and control software for the gas systems of TEC/TRD, DC/PC, MuID and MuTr detectors in PHENIX experiment, TPC detector in STAR experiment and test gas system of ATLAS CSC detector is described. In spite of gas systems difference, using of the object oriented model and multitasking environment make it possible to build a software package oriented to an abstract gas system. This software is easily adjusted for every mentioned gas system.

1 Introduction

The software described in this document was developed to control gas systems of STAR [1] and PHENIX [2] experiments at RHIC [3]. These gas systems were built for the following detectors: TEC/TRD, DC/PC, MuID and MuTr in PHENIX experiment and TPC in STAR. The same software was used for the ATLAS CSC detector test gas system. Every gas system has a dedicated computer for data acquisition and control. The software provides manual control of the gas system, data acquisition, shows measured system parameters and maintains gas system database. It also reacts on some gas system faults and takes care for automatic recovering.

2 Software requirements

During the gas system development the following software requirements were elaborated (Fig. 1):

- reliability and stable work without an operator,
- automatic handling of the gas system faults and alarms,
- friendly operator interface for the manual gas system control,
- all gas system parameters should be displayed,
- parameters history accumulation with fast access and revision,
- gas system events (alarms, software messages etc.) history accumulation,
- fast software adaptation to all gas systems (that use different electronics),
- fine tuning of the software by an operator (without the developer),
- remote access to the actual parameters of the system and actuating devices state,
- remote access to the parameters history.

The best way to comply these requirements is using the Object-Oriented Programming [4], particularly code reuse and component building. This enables the software to be universal and easily applied to different gas systems with minimal changes.

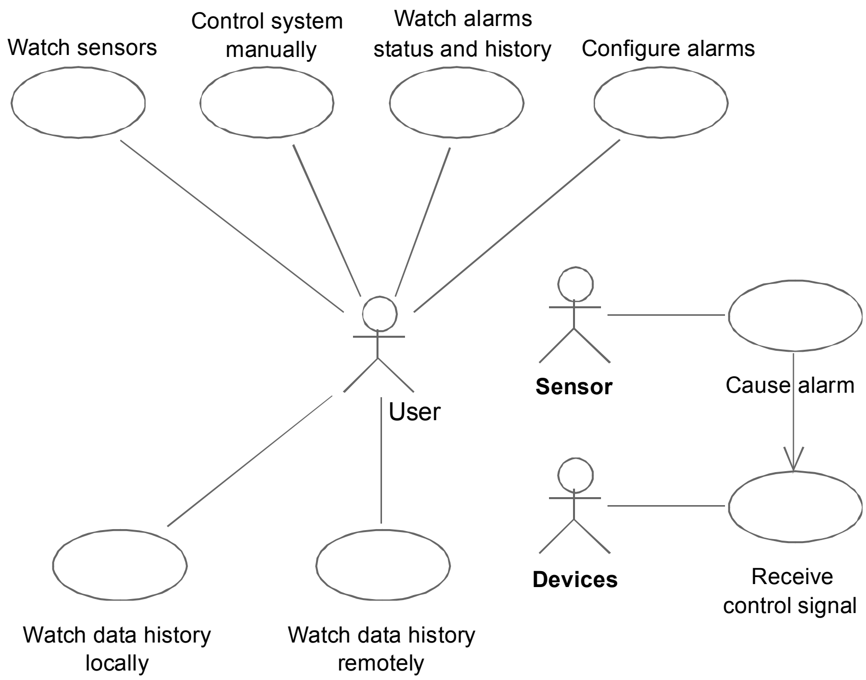

Fig. 1. Gas system software use case UML diagram

In order to improve software stability it is advisable to use multitasking environment with independent parallel processes. This allows to separate the most crucial part of the software, data acquisition, control and alarms handling, of potentially unreliable tasks like data saving and history access. Besides, for the parameters history accumulation it is reasonable to use database management system (DBMS). This speeds up data access and search and allows one to use standard software tools for it. The gas system control software was developed using Delphi for the Windows 2000/XP system.

3 Software structure

The gas system software consists of a number of parallel processes (Fig. 2). Gas system control process implements manual and automatic managing of the gas system parameters and devices. It reads out all sensors and controls actuating devices state via special hardware interface block, which is different for various gas systems. Sensors data are handled and put into the shared memory together with the timestamp of the reading. This procedure is detached to a separate thread (DAQ Thread) to improve data acquisition stability. The same thread takes care of the alarm conditions and corrective actions in the system. All gas system or software events are recorded to a special history file. Second thread (GUI thread) provides graphical user interface and responds to the operator actions.

The shared memory area is used by the “one writer, many readers” scheme [5] that simplifies process synchronization. After writing the parameters into the shared memory DAQ thread sets a system kernel event “GAS_DATA_READY”, which synchronizes other processes.

The Data Writer process reads all gas system parameters by the kernel event from the main process and stores them into MS Access database using COM [6] (Component Object Model). Separation of this procedure to independent process makes the control software much more reliable in case of database engine faults.

A third process is used to send gas system parameters to the central slow control system making use of the same kernel event. It is also separated in order to minimize influence of network operations to the main control process. Besides, this part of the software could be changed for different gas systems as it depends on the slow control system and it is reasonable to have a possibility to change it without modification of the main program. In STAR TPC gas system software all parameters are sent into EPICS [7] control system. There are two possibilities to send them: either writing a text file to the network drive, which is consequently read by EPICS, or direct parameters transfer to the EPICS database using EZCA library that is ported to Windows.

A separate program, DB Viewer, provides fast data history access and visualization using MS Access database engine. Remote access to the actual gas system parameters and database is enabled by custom WWW server. It handles user queries made by web browser, translates it into SQL[1] queries [8] and transfers acquired data back to the user as a chart or text file.

[1] Structured Query Language

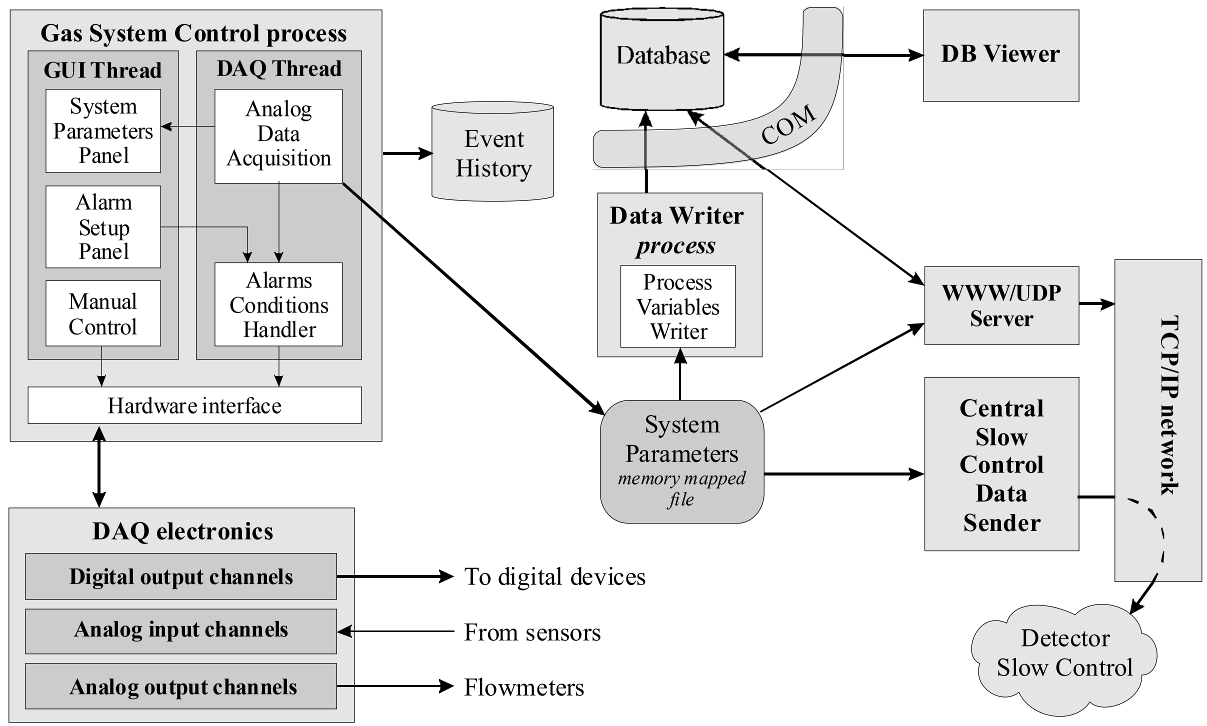

Fig. 2. Gas system control software structure.

3.1 Gas system control process

The main control process consists of two parallel threads (Fig. 2). First thread provides graphical user interface, while second thread implements data acquisition and gas system control. This separation allows to avoid visual interface delays during the data reading and makes more reliable alarms handling. However, some difficulties appear connected with threads synchronization [5]. Particularly, any electronics access is initiated in DAQ thread in order to eliminate protocol collisions. An operator works with GUI thread, which sends a signal to the DAQ thread in case of manual control action (Fig. 3). DAQ thread consequently communicates with electronics.

DAQ thread has maximal priority (Time Critical) in the process priority class which is highest possible thread priority for all process priority classes except real-time processes [9]. Hence the software has fast and stable reaction time for the gas system events independently of the operator actions.

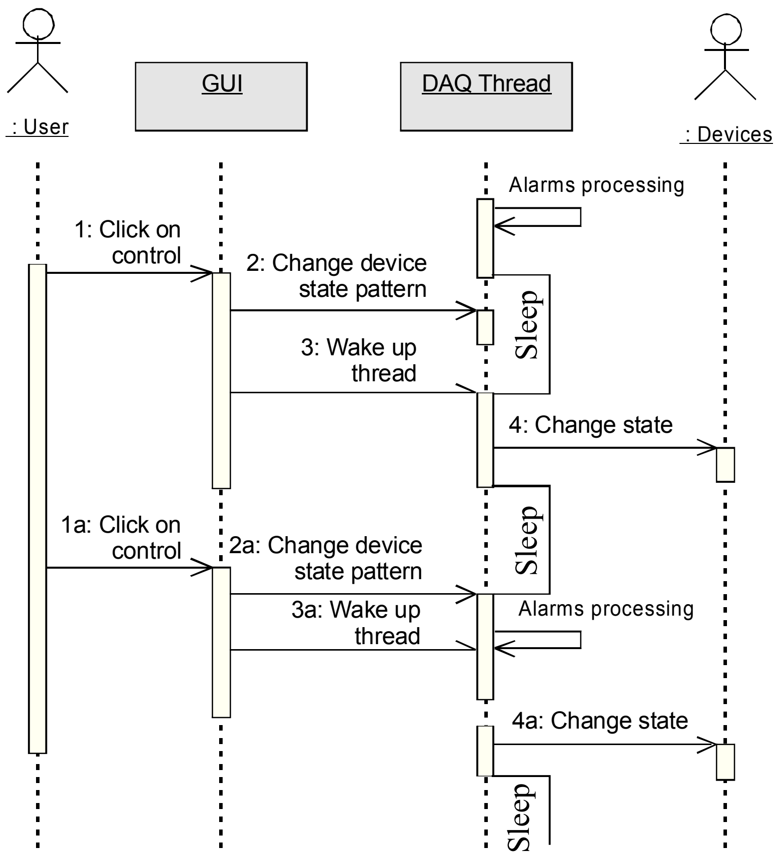

Fig. 3. Manual control sequence diagram

Sometimes DAQ thread could be in the idle state. This is usual behaviour between the data acquisition cycles. In case of GUI thread event, notifying about operator actions, DAQ thread wakes up, performs necessary control actions and sleeps again (Fig. 3, steps 1-4). Idle time is calculated every time to ensure the constant total idle time between data reading cycles. If GUI thread signal comes during the data acquisition or alarms handling (steps 1a-4a), DAQ thread finishes data handling cycle and only after that performs control actions. This allows to avoid electronics exchange conflicts.

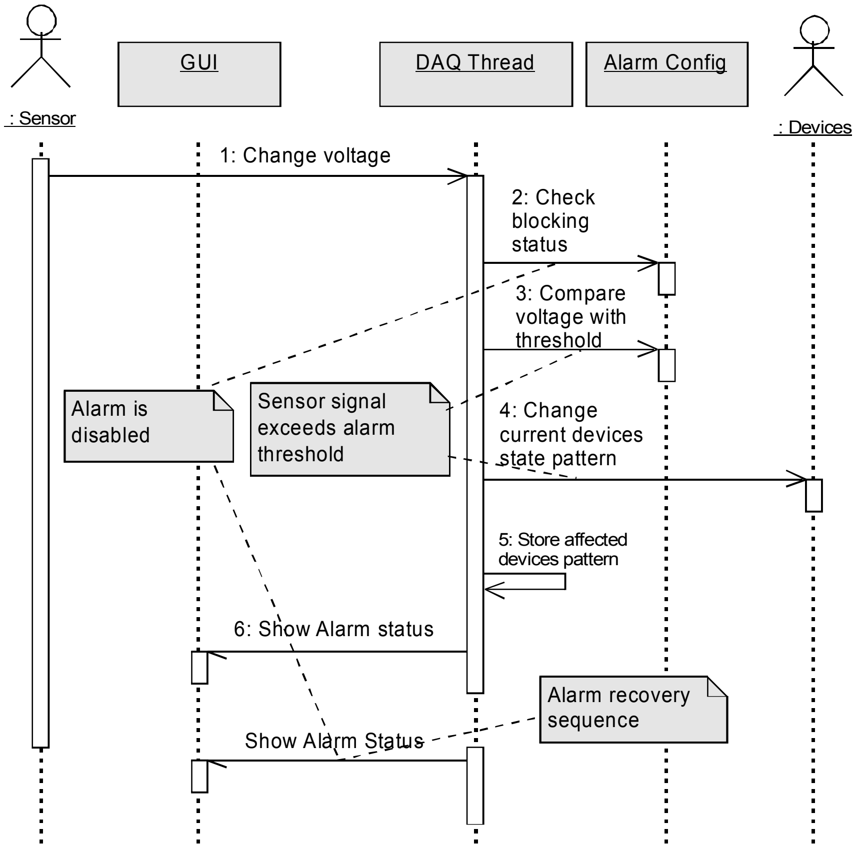

Fig. 4. Alarm handling sequence diagram

An alarms processing procedure in the DAQ thread is very similar to that used in the Alarm/Interlock device [10]. After every data acquisition cycle sensor signal is compared to the corresponding threshold (Fig. 4). If sensor value exceeds the threshold, program takes corrective actions according to that alarm configuration and remembers the actuating devices affected in this procedure. This information is used later for the alarm recovery procedure (Fig. 5). If sensor signal comes back to the normal range (below the threshold), DAQ thread acts only on the devices that are not affected by other alarm procedures. Real devices state is changed only after all alarms processing.

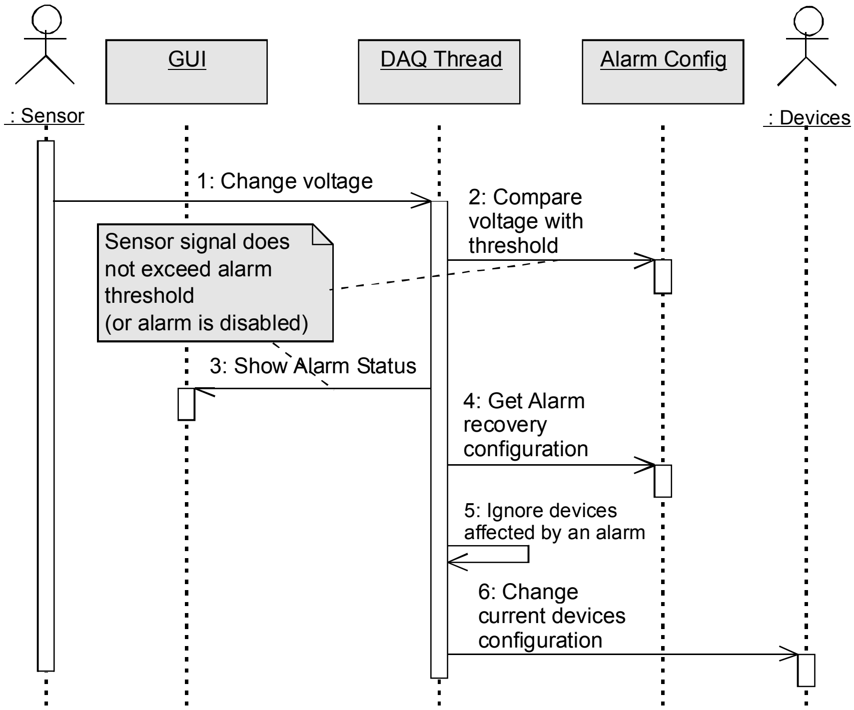

Fig. 5. Alarm recovery sequence diagram

Alarms configuration is similar to the Alarm/Interlock device settings [10]. In contrast to the device configuration, the number of alarms is not limited in computer software. This enables multiple alarm settings (high, low, warning and alarm levels) for the single sensor. There is also *count* parameter, which was first implemented in STAR TPC gas system software. This parameter defines the number of sensor samples exceeding the threshold, necessary to trigger the alarm.

The data acquisition and handling period is set by an operator and usually is 60 seconds. The data is written into the database immediately after the alarms processing. In the MuTr gas system software a second period was first implemented – database writing period. The MuTr gas system is not equipped [11] by the Alarm/Interlock device, but relatively fast DAQ electronics is used instead (National Instruments SCXI). The gas system software reads data and process alarms rapidly. But it's not necessary to write the data into the database so often, because usual gas system reaction takes minutes. This is solved by implementing two operator controlled periods: data reading (1-10 second) and database writing (60 seconds and more).

All software settings could be exported into a registry text file. Besides, alarms configuration could be exported into an Excel report, as a map of actuating device states in different conditions. Alarms are shown in rows and control devices in columns.

The Gas system controller main window provides a manual system control (Fig. 6). All actuating devices are displayed in a symbolic form on the simplified gas system scheme. Device symbols are active, mouse click on any device symbol changes corresponding device state. They are represented by a special object which simplifies reuse of the device symbols in different gas systems. All gas system parameters, sensor readings and calculated values, are shown in the right part of this window. There is also an event log in the bottom showing all alarms or software messages.

3.2 Data storage and handling

The gas system parameters are read by the Gas system control process and put into the shared memory area. This data is then read by other processes by system kernel event signal. Thus, shared memory always contains actual gas system parameters. A Microsoft Access database is used for the data storage. All database operations are performed in the form of SQL queries [8]. The Data Writer process reads all gas system parameters from the shared memory, generates and run a SQL-query to insert them into the database. The database engine does all low level operations like data search, sorting etc. This makes data access much faster and simpler.

The gas system is intended for long working period without an operator intervention. Hence the database should be designed to keep all system parameters for at least 6 months period. The data writing and selecting speed must not essentially decrease with database growth. A special program was written to test database engine. The test showed that consecutive insert of 400000 records doesn't affect the insertion time, which was measured to be 10 ms on the Pentium 200 MHz PC. This is absolutely sufficient for the reliable work of the gas system software. This number of records corresponds to more than 9 months of the software work. It is reasonable to start a new database for every new gas system run. This feature is implemented in the Data Writer.

The Data Writer program was designed for different gas system software. Every database contains a special table with gas system parameters definition, while the data history is kept in a separate table. The Data Writer uses main control process settings to fill the gas system description table during the database creation. The data history table is the same for all gas systems. This makes the Data Writer process universal and independent on the gas system.

A special monitoring software, DB Viewer, was developed for data selection and visualization. It works with the database using COM interface [6]. This program generates SQL-queries on the base of starting date, time period (day, week or month) and gas system parameter, chosen by an operator (Fig. 7). Usually the DB Viewer is used for daily monitoring of the actual gas system parameters. User-friendly interface allows one to zoom or move the chart, save it in WMF (windows metafile) format or as a text file for further analysis. Data is averaged by a minute (for daily time period) and by an hour (for weekly and monthly periods).

In addition, the DB Viewer software makes it possible to get data selection as a table (Fig. 8). This selection could contain several gas system parameters for a chosen time period (1 day increment). User could also choose the averaging mode: hour, day or no averaging. Result table can be exported as a text file or MS Excel spreadsheet for further analysis. The DB Viewer has also some service features for the database support. There is a possibility to get a statistical report of the database, including number of records a day, delete unnecessary data (e.g. during gas system tuning) and compact the database (physically erase deleted data).

The DB Viewer program was designed to be used in all gas systems mentioned above. It reads system-dependent information from the database. The gas system databases can be transferred easily. To use the gas system data one need only Microsoft Access engine, that is included in standard Microsoft Office and in the latest Windows systems.

3.3 Remote data access

During the gas system run sometimes it is useful to get an expert supervision. Remote data access provides this feature for the distant experts. A special program, WWW/UDP Server, was developed for the following purposes:

- parameters history (database) access in the form of charts or data files,
- actual gas system parameters monitoring,
- actuating devices state display.

All these tasks could be realized using any standard web-browser on the client side. Server obtains a user request, generates a SQL-query and transfers required parameter chart as a web page back to the browser (Fig. 9), similar to the DB Viewer program. Operator can download a number of gas system parameters in a text file, like in table query in the DB Viewer.

The server software also reads actual gas system parameters from the shared memory area by user request and sends them back to the users' browser as a web page. The control device states are transferred as a picture, similar to the manual control window in the main control program.

Remote control of the gas system is denied through the web server and through the detector slow control system as well.

3.4 Software adaptation

The gas system software was developed to control and monitor of the various gas systems. The differences of those systems are DAQ electronics, sensor and valves parameters and alarms configuration. In PHENIX experiment, TEC/TRD and DC/PC detector gas systems are using Keithley digital multimeter and switch system for the data acquisition. The gas systems for MuID and MuTr detectors in PHENIX experiment and TPC detector in STAR equipped by National Instruments SCXI electronics. The gas system for ATLAS CSC detector test bench makes use of Alarm/Interlock device [10] for the sensors readout and control.

The software concepts are the same for all systems. This similarity, together with the object oriented model use, make it possible to build a software package oriented to an abstract gas system. The gas system software adaptation to the specific system includes several tasks:

- electronics binding (creation of the DAQ electronics interface for the main control process),
- simplified gas scheme with control devices display,
- sensor parameters configuration (names, units of measurement, voltage to value conversion coefficients),
- alarms configuration (alarm thresholds, actuating devices maps) – elaborated during gas system development and tuning.

These tasks usually take about one or two working days. The Data Writer process automatically takes all necessary information about the gas system from the main control program configuration. The DB Viewer program is independent of the gas system, because it reads all required information from the database, as well as the WWW/UDP Server does.

4 Experimental results

The gas system software is now installed and used on 6 gas systems at BNL (Brookhaven National Laboratory, USA). During 4 years operation it showed high stability of its main tasks – system control and monitoring. There were no serious faults, which could affect the gas system operation.

In STAR TPC gas system some sensors showed unstable readings. This was fixed by the *count* parameter in the corresponding alarms configuration. One minor fault was observed in the Data Writer program, connected with the lack of memory. It didn't affect the main control program, which is one more proof of the multitasking profit. After computer upgrade this problem doesn't show up anymore. All database-related programs also performed good and stable. Visualization software showed high usability.

Flexible software configuration allows an operator to change and add alarm handling procedures easily, as gas system operation experience increases. Besides, similarity of the software in all gas systems simplifies operator's work dramatically.

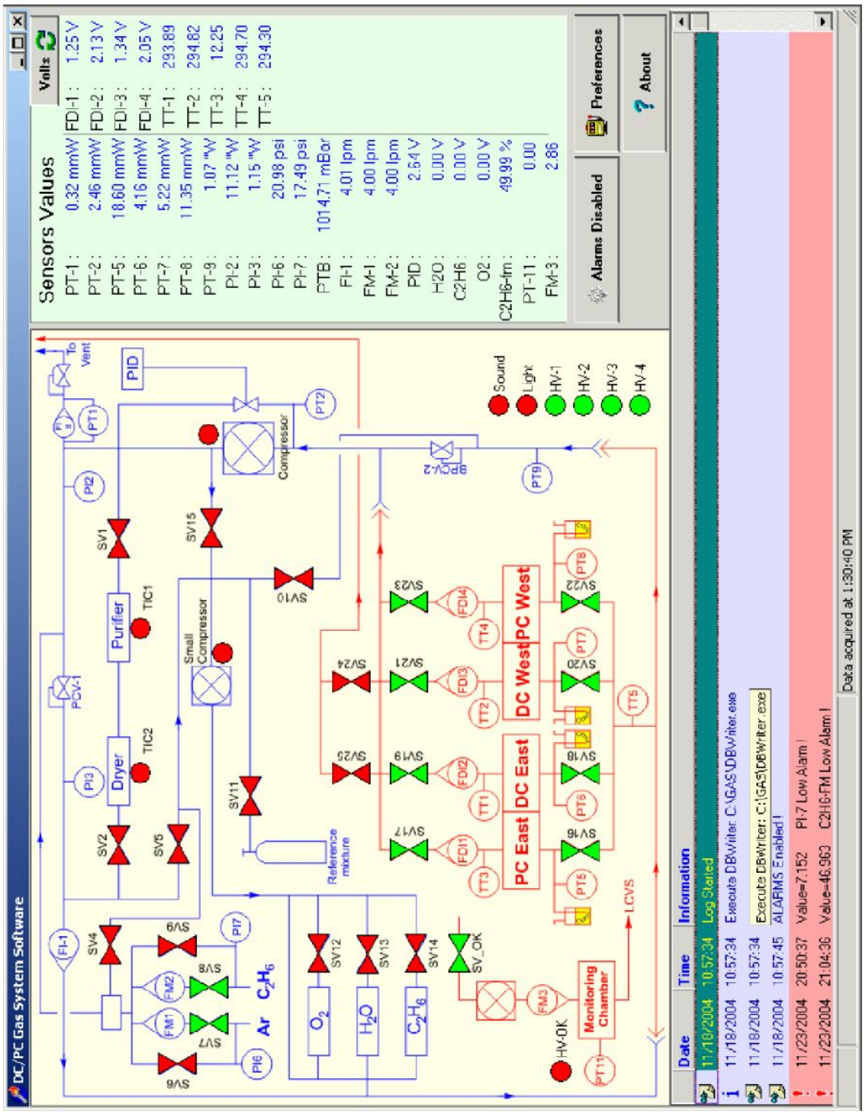

Fig. 6. Main window of the Gas system control process.

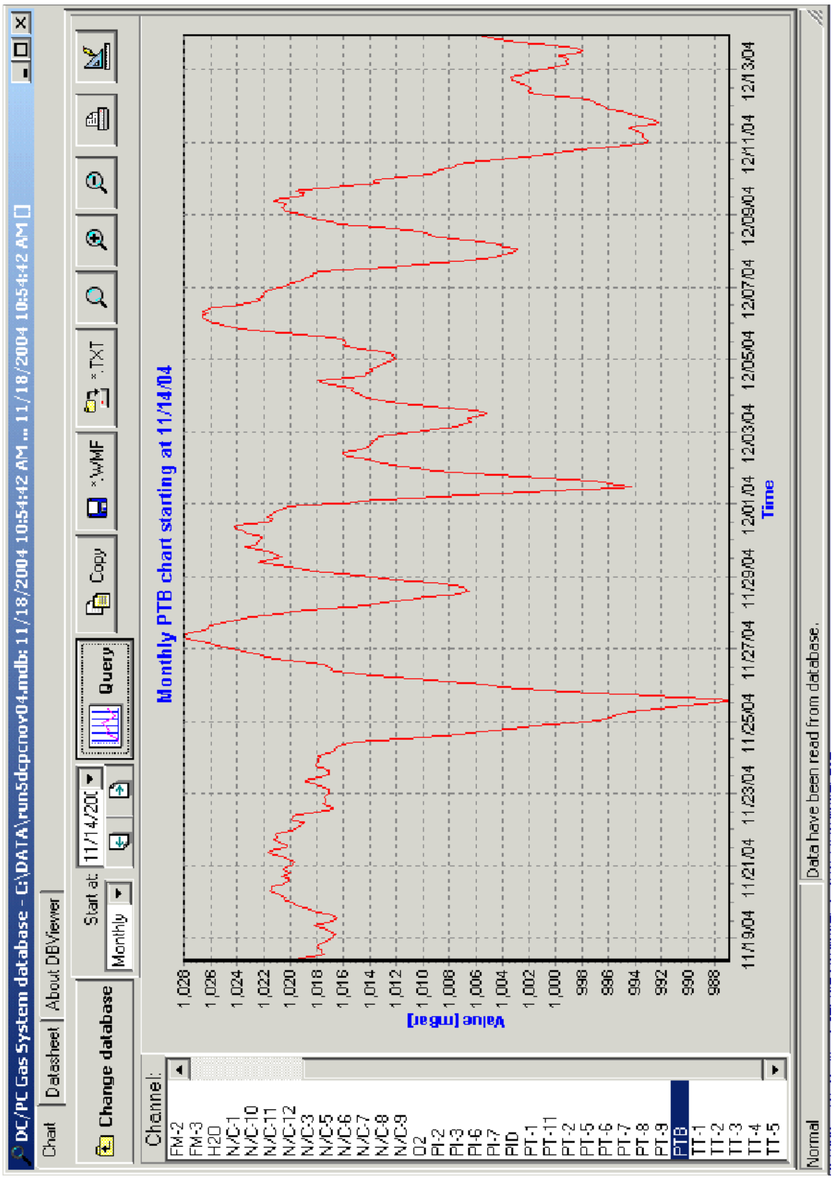

Fig. 7. Gas system parameters visualization in DB Viewer.

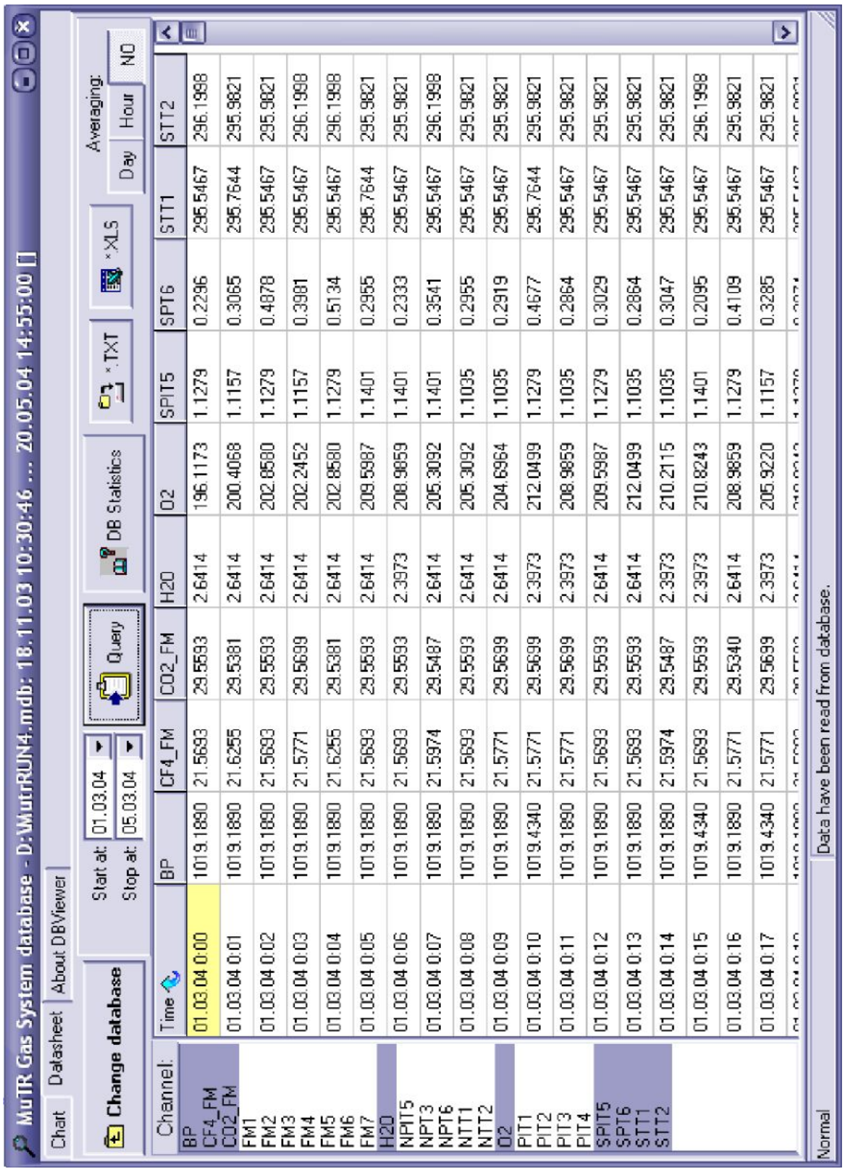

Fig. 8. Table gas system data representation.

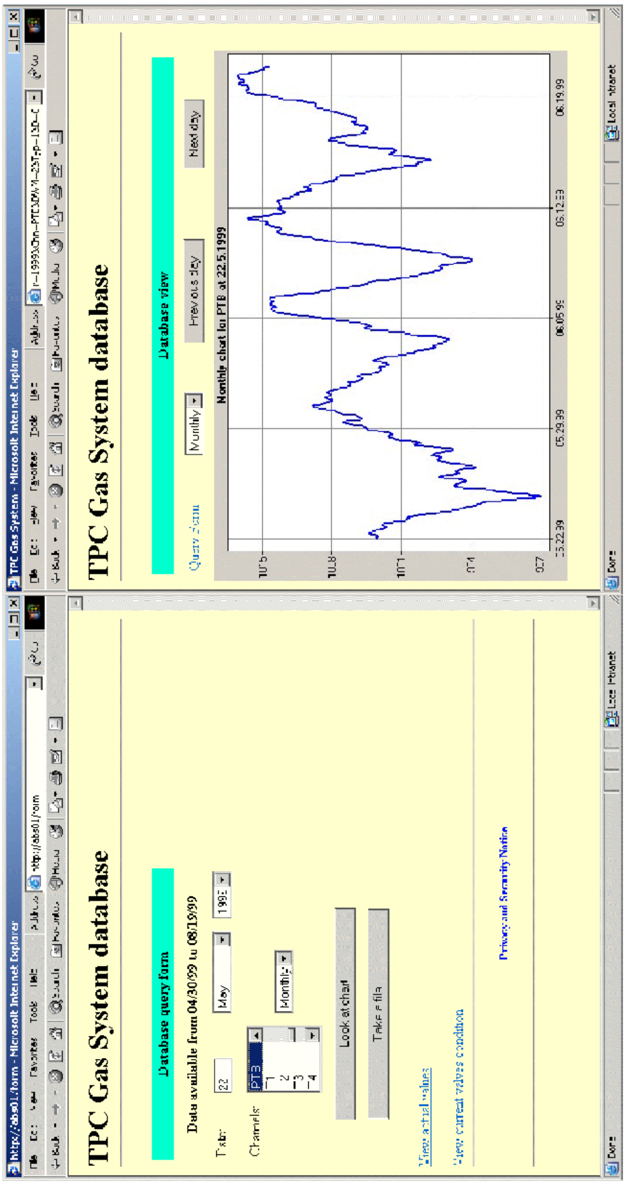

Fig. 9. Remote access to the gas system data using web-browser.

References

[1] K.H. Ackermann, N. Adams, C. Adler *et. al.*, NIM **A 499** (2003) 624-632.

[2] K. Adcox, S.S. Adler, M. Aizam *et. al.*, NIM **A 499** (2003) 669-679.

[3] M. Harrison, T. Ludlam, S. Ozaki, NIM **A 499** (2003) 235-244.

[4] G. Booch, *Object-Oriented Analysis and Design with Applications. 2nd edition.* Addison-Wesley, 1994.

[5] A. Cohen, M. Woodring, *Win32 Multithreaded Programming.* O'Reilly, 1998.

[6] D. Rogerson, *Inside COM*. Microsoft Press, 1997.

[7] A.J. Kozubal *et al.*, *Experimental Physics and Industrial Control System*. ICALEPCS89 Conference Proceedings, Vancouver, 1989, 288.

[8] M. Gruber, *Understanding SQL*. Sybex Inc, 1990.

[9] J. Richter, *Advanced Windows. 3rd edition*. Microsoft Press, 1997.

[10] L. Kotchenda, P. Kravtsov, V. Trofimov, *Universal microprocessor interlock system*. Instruments and Experimental Techniques **43**, N 4 (2000), pp. 504-505.

[11] L. Kotchenda, P. Kravtsov, R. Pisani *et. al.*, *PHENIX Muon Tracking Detector Gas System*. To be published in NIM.