DEVELOPMENT AND IMPLEMENTATION OF SYSTEM SOFTWARE COMPONENTS FOR ATLAS EXPERIMENT AT CERN

Yu.F.Ryabov

1. Introduction

The paper briefly presents the main directions of the informatics laboratory (LIVS) activities in the ATLAS experiment, being prepared on the LHC accelerator at CERN. The results have been achieved by several working teams, which the participants of the presented job belong to as basic developers.

The following two chapters correspond to two big domains of the activities:

1) **ATLAS TDAQ [1]/DCS [2] project** including
   - ATLAS DAQ Online software [3] [1] – a series of software subsystems providing the configuration, control and interaction of processes in DAQ. The laboratory members develop configuration database, inter process communications, monitoring and diagnostics components and participated in run control and graphical user interface design. Beginning time of working for the Online software – 1993.
   - Detector control system: general DCS architecture, configuration database, input data from the CAN field bus devices (since 1995).
   - Communications between DCS and DAQ (since 2000).
2) **Control systems for production of TRT detector of ATLAS** including
   - The database and interactive tools with the Internet access for quality control of the wheel production for TRT (since 1998).
   - Slow control systems for the TRT production area environment control and for the wheel test station (since 1997).

All the developments presented are based on the recent advanced technologies like object software design and implementation, common object request broker architecture (CORBA) [2], object database and knowledge base technologies, recent network technologies (field buses, LAN, WAN), HTTP on-fly requests and industrial SCADA [4] products.

2. Program Components of DAQ and DCS of ATLAS Experiment

The design and integration of the components described below into the ATLAS TDAQ/DCS are included into the Technical Proposal document [5]. The set of DAQ/DCS system software components is shown in Fig. 1, where the gray background highlights the components being developed with the PNPI participation.

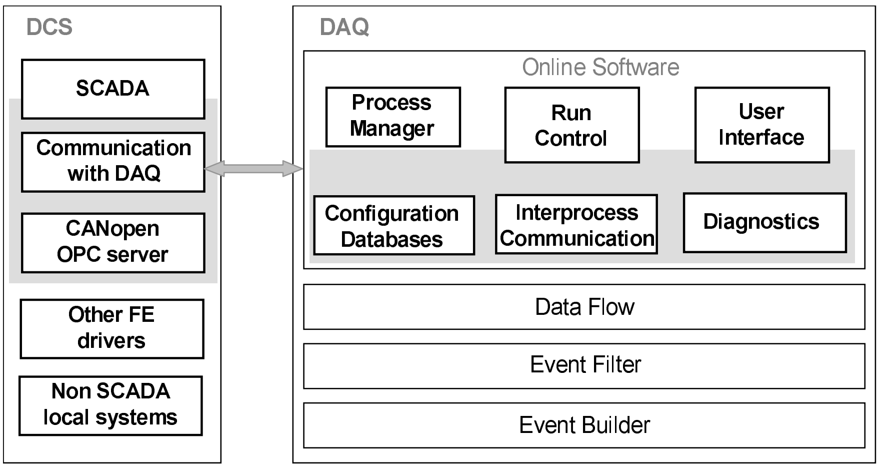

Fig. 1. ATLAS DAQ/DCS Software components
(Gray background highlights PNPI participation)

[1] Trigger and Data Acquisition.
[2] Detector Control System.
[3] formerly known as the Back-end software.
[4] Surveillance Control And Data Acquisition.
[5] ATLAS high level Triggers, DAQ and DCS Technical Proposal, CERN/LHCC/2000-17, http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/SG/TP/draft_tp.html.

2.1. DAQ Online Software

2.1.1. Configuration Databases

The data acquisition system needs a large number of parameters to describe its system architecture, hardware and software components, running modes and the system running status. One of the major design issues of ATLAS DAQ is to be as flexible as possible, externally parenthesized by databases. The ATLAS DAQ Online software configuration databases [3, 4] are the following:

- Hardware configuration database;
- Run control configuration database;
- Software configuration database;
- Data flow configuration database;
- General run parameters configuration database;
- Detector parameters configuration database.

In order to keep and manage the databases during the preparing and performing the data acquisition periods (DAQ runs) the OKS (persistent in-memory object manager) [5] has been developed, providing major functionality of a distributed object database management system. The schema of the configuration databases is designed and generated using StP/OMT tool. A problem-oriented data access library for the remote access of the DAQ program applications for the configuration databases is implemented [6]. The results of the configuration management tools showed that the current implementation is reliable across the DAQ Online software operating systems and has good performance [7].

2.1.2. Interprocess Communication Facility

ATLAS DAQ should provide the possibility of performing several independent runs, *i.e.,* run simultaneously several instances of the same software components in different contexts (e.g., on different sub-detectors [6]). In accordance of this requirement, the partitioning concept has been followed while developing the on-line software [1]. In order to allow that, the inter process communication package (IPC) [8] has been developed. The IPC package provides also a unified way of communication between the software objects simplifying the development of component to be available for each other and for the users on a network. The package is built on top of the ILU implementation of CORBA [9]. It provides standardized server and client application program interfaces hiding behind them the technical details of CORBA.

Most of the Online software including the following core components have been developed with making use of the IPC package.

- Information service (IS) [10]

  The main goal of IS is to provide a mean of information exchange in the distributed environment. It provides for the program applications running at the same or different partitions the client interface both to publish data and to receive information from IS when it has been changed or on request of a receiver.

- Message reporting system (MRS) [11]

  The MRS facility gives the means for the reporting and the reception of unique text information (*e.g.*, error messages). A DAQ application can report its messages for any partition and subscribe in any partition MRS server for the messages that satisfy a certain criteria defined on a combination of the message descriptors.

- Run control (RC) [12]

[6] this especially important for the test beam runs.

The RC component controls data taking activities by coordinating the operation of the DAQ sub-systems, Online software and external systems. It has user interfaces for the shift operators to control and supervise the data-taking session and program interfaces for the DAQ sub-systems and Online software components.

2.1.3. Diagnostic Verification Component

The diagnostics system (DS) [13] is the components of the ATLAS Online software to assist a DAQ operator to initialize, test, setup and run the DAQ system without deep knowledge in its structure and functional features. The DS will be implemented in two separate components:

- The verification component of the DS (DS VC) uses the test manager (TM) to test the DAQ configuration and confirm functionality of any DAQ subsystem. By grouping tests into logical sequences, the DS can examine components of the system (hardware or software) at different levels of detail in order to determine as accurately as possible the functional state either of a component or of the entire system.
- The supervision component of the DS (DS SC) helps the DAQ supervisor to control DAQ, diagnose and recover DAQ errors during setup, run and shutdown phases in the automatic or the operator assistance modes.

The verification component is to be used mainly before and during the DAQ booting and setup, but it may also be used by the DS SC during DAQ run in the case of DAQ fault if the DAQ supervisor or DS SC decides that some testing is required. DS VC works in global partition context like TM and the process manager do. Thus, it is able to test any component in any partition.

The heart of the VC is an expert system (ES) – CLIPS [14] is chosen for the prototype [7] that consists of:

- Knowledge base that holds all specific knowledge (rules) how to test a hierarchy of DAQ components.
- Working area that is dynamically updated and reflects the DAQ current status inside the ES as well as the progress of testing.
- Inference engine that executes rules and updates the working area.

2.2. Detector Control System

The architecture of the distributed ATLAS detector control system [8] includes a series of the SCADA based applications (for sub-detectors and top-level supervisory control) and front-end systems. For the latter, a general-purpose front-end electronic module (LMB, ELMB) [16] and an OPC server (see 2.2.2) to drive sets of them have been developed.

2.2.1. Configuration Database

Like it has been told above for DAQ, the large distributed detector control system needs the hardware and software components, their parameters, etc., to be described and stored in an external configuration database. This is necessary both for the inquiries while the DCS maintenance and for the downloading run-time parameters into the front-end systems (and perhaps into the SCADA run time database).

An object model of top layer DCS description in the configuration database has been developed [17, 18]. The model includes the object classes for a hierarchical decomposition of DCS beginning from sub-detector control system down to hardware and software elements and defines considerable relationships between them. The concept of Control_Unit object class defining an elementary entity of the system functional structure is proposed [17]. A Control_Unit object integrates a series of hardware, software and DCS parameter objects related to a certain elementary [9] DCS control function (e.g., control of a power supply).

---

[7] though the DS design is independent from any expert system specific technology.
[8] where the PNPI proposal [15] has been taken into account.
[9] *i.e.*, that is not a composition of any other "inner" functions in the model used. Obviously, the definition of "elementary" functions is model-dependent.

A way of model definition for the sub-detector control developers is suggested [17] and a notation scheme of the system components (objects) is proposed [18].

2.2.2. CANopen OPC Server

The destination of the OPC CANopen server [19, 20] is the control and data acquisition of devices based on the CAN field bus and using CANopen protocol for the communication. The server has the interface defined by the standard "OPC Data Custom Interface 2.0". For the time being, practically, all the SCADA industrial products including PVSS II system [10], having been chosen as the basic software product for all control systems of LHC experiments, make use of this standard. The OPC standard is based on the COM/DCOM technology of Microsoft Company destined for the developing program components of distributed systems.

The server is capable to drive CANopen-devices of different vendors, in particular, the LMB/ELMB modules [16][11] that are the basic front-end electronic modules for the ATLAS detector control.

In order to support easily the driver boards of different vendors, the server is composed of two parts. The first one implements all the interfaces how they are specified by the OPC standard while the other performs the data acquisition and control of CANopen devices. This considerably simplifies the developing OPC servers for different boards.

The server has been used for the tests of LMB [21] and is being used at several CERN experiments and tests beams now.

2.3. DAQ – DCS Communication

The DAQ–DCS Communication (DDC) software of the ATLAS TDAQ/DCS system is intended to be an interface between DAQ and DCS and to support all needs of run time information exchange [21, 22]. The domain decomposition defined 3 functions the DDC shall provide:

1) Bi-directional exchange of data like parameters and status values.
2) Transmission of DCS messages, like alarms, to DAQ.
3) Ability for DAQ to issue commands on DCS.

The functions listed above are independent and are being implemented as separate subsystems. The following prerequisites are used while the design and implementation:

- The Online software [1] is the interface point for the DDC on the DAQ side (they are IS, MRS and RC subsystems in the order of the DDC function listed above).
- The DCS is implemented by a SCADA system. The PVSS II SCADA product is utilized for that purpose and its API is used to connect the DDC.

The prototype of the DDC package included into the Online software release 0.0.14 (May 2001) provides the functions of the data exchange between DAQ and DCS as well the transferring of DAQ commands for DCS and giving the feedback [12].

3. Software for TRT Production Management and Control

3.1. Production Database of ATLAS/TRT Detector

The production database of ATLAS/TRT detector is developed to provide information support for TRT end-cup wheels production and assembling [23]. The main functions of this information system:

- control of the consistency of the technological operations' execution,
- quantity control on each stage of wheel production,
- storage of tests for quality control during the wheel production,

[10] see http://www.pvss.com
[11] see also http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/ELMB/elmb.html
[12] see http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/ddc.html for details

- storage of the raw test data,
- online analysis of data dependencies,
- access to needed data via Internet.

All the data are collected through a local network of workstations, which are installed with MS Access97 as a front-end user interface and a local database. The main database server, which is based on ORDBMS [13] Oracle8, is used to store the bulk of the data and provides straightforward access to the data via the Internet. The logical structure of the system is shown in Fig. 2.

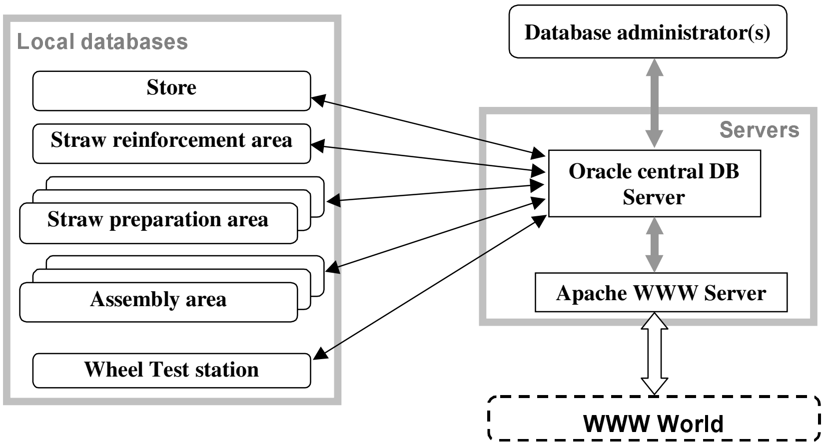

Fig. 2. Logical structure of ATLAS/TRT information system

Production and assembly operations are grouped to several groups referring to certain production areas.

Local Databases (LDB) are created and maintained on each workstation. All LDB have the same structure and software tools. The difference between workstations is defined by the set of the technological operations that are performed at the workstation. All local databases are able to work independent of each other and of the central server.

At the local workstation all data are accessible through MS Access interface. To simplify the remote access for the data and to give a possibility of creation of conditional request, the Web site for interactive access for the data stored in central Oracle database is developed (http:\\atlas.pnpi.nw.ru\dbase.htm). This site presents the following information about the wheel production:

- The current state of produced component and received materials for each production area.
- The production rate for each kind of production per month or per days.
- Graphic representation of plan and of real amount of the components produced.
- The material consumption.
- Detailed information for each selected batch of material or of produced components.
- Graphic representation of distribution of produced component by categories.
- Interactive online analysis of dependencies between stored data (cross queries).
- Test results and their correlation with the previous data.

An original method has been used to create the graphics "on-fly" as the result of a dynamic request processed by CGI program, but not from the pictures prepared beforehand.

Additionally, the e-mail report generator is developed that allows to send to subscribed users periodic (weekly/monthly) reports with the brief description of the produced components and the reports about the consumption of materials used for production.

[13] object-relational database management system.

3.2. ATLAS/TRT End-cap Wheel Test Station Databases

The separate databases were developed for Wheel Test Station (WTS). The main goal of WTS is to check performance of all wheels channels under working conditions, in particularly to measure the straightness of the straws and position of the wires in the wheel. Procedure of end-cap wheel testing consists of the following stages:

- definition of hardware configuration;
- calibration of electronic channels;
- test measurement run;
- express data analysis.

Data collected and calculated during data acquisition and data processing are stored as work files on the workstation, controlling the test run. In order to guarantee the reliable data storage, data of measurements and calculated parameters are stored for the long-term use into Oracle database and, simultaneously, in the File database created at the shared area of disk storage that is accessible from user workstations through LAN (Local Area Network). Oracle database will be used mostly for the Internet access and LAN users will use the File database.

The Web site for interactive access to the data of WTS is under realization and will support the next main functions:

- Graphical and tabular representation of the wheel test results.
- Graphical and tabular representation of the parameters that correspond to history of straw production and wheel assembly (so named ***context***).
- Sending by request the text comma-delimited file that contains the calculated data for restricted number of selected straw or context information for restricted number of selected straw.

3.3. Slow Control System for TRT Production Area Environment and Wheel Test Station

The main purposes of the slow control systems for the TRT production area and for the wheel test stations are the following:

- Monitoring the atmospheric parameters at the TRT assembly area: temperature and humidity.
- Monitoring the atmospheric parameters at the wheel test station area: temperature, humidity and pressure.
- High voltage and current control of the wheel test station.
- Motion system for wheel test station.

The system structure, which has been developed for the slow control system, is shown in Fig. 3. It includes:

- PC-compatible computer with Bridge View 2.1 software.
- Local Monitor Box (LMB) for temperature measurement.
- Humidity and pressure measurement device.
- Humidity measurement device.
- Motion system.
- NYM High voltage power supply module.

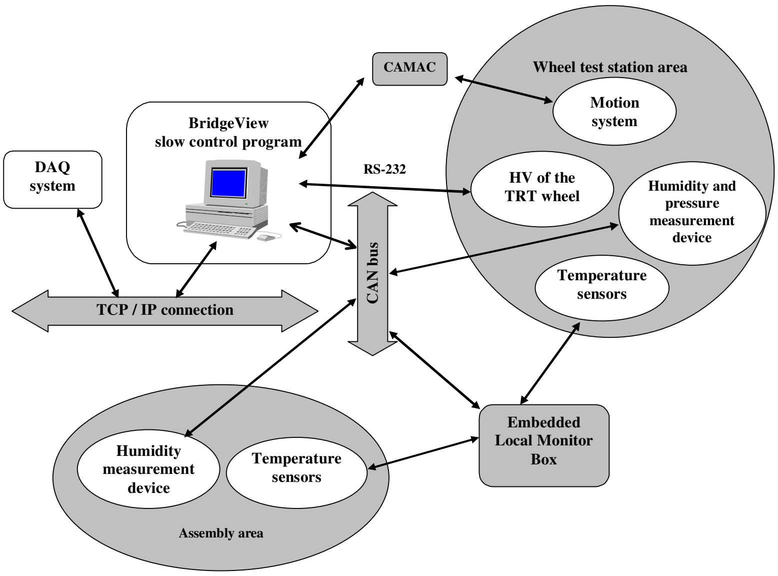

Fig. 3. General structure of TRT wheel production slow control system

The following electronic devices have been developed for that slow control system:

- Humidity and pressure measurement device.
- Humidity measurement device.
- Motion control module.

The system software is developed on the base of Bridge View 2.1. The structure of software is shown in Fig. 4.

The main program provides the hardware control, monitoring of current parameters (including Internet access) and storing them into an internal database. It is responsible also for the communication with DAQ. The historical browser allows looking through all the data kept earlier. In further this service will be also accessible via the Internet.

The system of TRT wheel production slow control works at CERN since April 2001. Similar systems will be produced for PNPI and JINR (Dubna).

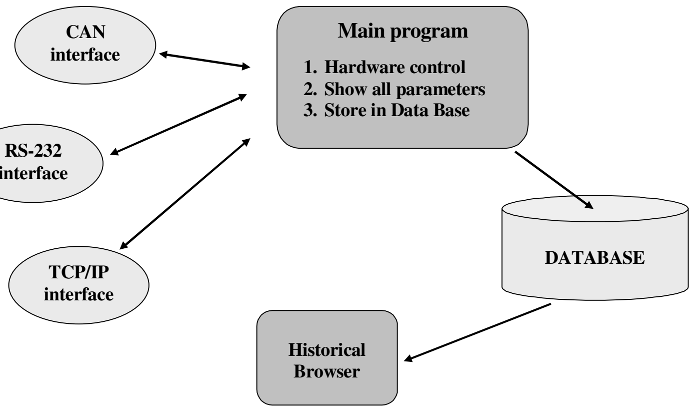

Fig. 4. Software structure of Wheel Test Station slow control

4. List of participants

The list of the participants of the job presented in the paper: V.M.Filimonov, S.V.Katounine, A.G.Kazarov, V.P.Khomoutnikov, N.V.Klopov, S.E.Kolos, E.G.Novodvorsky, S.B.Olechko, Yu.F.Ryabov, and I.B.Soloviev.

References

[1] I.Alexandrov, … , S.Kolos, Yu.Ryabov, I.Soloviev *et al.*, in *Proceedings of the International Conference on Computing in High Energy Physics (CHEP 1998), Chicago, USA,* 1998.

[2] S.Kolos, in *Proceedings of CERN School of Computing, Marathon, Greece, 2000*, p. 133.

[3] R.Jones and I.Soloviev, in *Proceedings of the International Conference on Computing in High Energy Physics (CHEP 1998), Chicago, USA,* 1998.

[4] I.Soloviev, DAQ Note –135, Sep. 2000; http://atddoc.cern.ch/Atlas/Notes/135/confdb-ug.html.

[5] R.Jones, L.Mapelli, Yu.Ryabov and I.Soloviev, IEEE Trans. on Nuclear Science **45,** 1958 (1998).

[6] S.Kolos and I.Soloviev, DAQ Note-122, June 1997.

[7] I.Soloviev, DAQ Note-114, Feb. 1999; http://atddoc.cern.ch/Atlas/Notes/122/Note122-1.html.

[8] S.Kolos, DAQ Note-75, Mar. 2000; http://atddoc.cern.ch/Atlas/Notes/075/Note075-1.html.

[9] R.Jones, S.Kolos, L.Mapelli, and Yu.Ryabov, IEEE Trans. on Nuclear Science **47**, 331 (2000).

[10] S.Kolos, DAQ Note-037, Aug. 1999; http://atddoc.cern.ch/Atlas/Notes/037/Note037-1.html.

[11] D.Burckhart, M.Caprini, R.Jones, S.Kolos, and A.Radu, DAQ Note-059, Feb. 1999; http://atddoc.cern.ch/Atlas/Notes/059/Note059-1.html.

[12] P.-Y.Duval, … , S.Kolos *et al*, DAQ Note-107, July 2000; http://atddoc.cern.ch/Atlas/Notes/107/Note107-1.html.

[13] R.Jones, A.Kazarov, S.Kolos, and Yu.Ryabov, DAQ Note-106, Apr. 1999; http://atddoc.cern.ch/Atlas/Notes/106/Note106-1.html.

[14] A.Kazarov and Yu.Ryabov, DAQ Note-108, Dec. 1998; http://atddoc.cern.ch/Atlas/Notes/108/Note108-1.html.

[15] V.Filimonov, V.Khomoutnikov, and Yu.Ryabov, ATLAS Internal Note DAQ-NO-043, Feb. 1995; http://atlasinfo.cern.ch/ATLAS/DAQTRIG/DCS/note43.ps.

[16] B.Hallgren, … , V.Filimonov, Yu.Ryabov *et al.,* in *Trans. of the International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS), Trieste, Italy, 4–8 Oct. 1999*, p. 199; http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/mc1p10.pdf.

[17] V.Filimonov, V.Khomoutnikov, and Yu.Ryabov, ATLAS DCS Internal Note DCS-IWN2, May 1997; http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/note0597.ps.

[18] V.Khomoutnikov and Yu.Ryabov, ATLAS DCS Internal Note DCS-IWN4, Dec. 1997; http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/note1297.ps.

[19] V.Filimonov, "OPC CANopen server", http://atlasinfo.cern.ch/ATLAS/ GROUPS/DAQTRIG/DCS/LMB/ELMB/DISTRIBUTION/opccano5.pdf.

[20] V.Filimonov and B.Hallgren, ATLAS DCS Internal Note DCS-IWN6, Dec. 1998; http://atlasinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/DCS/LMB/POOL/dcsiwn6.pdf.

[21] R.Hart and V.Khomoutnikov, DAQ Note-165, May. 2001; http://atddoc.cern.ch/Atlas/Notes/165/Note165.pdf.

[22] H.Burckhart, R.Hart, R.Jones, V.Khomoutnikov, and Yu.Ryabov, in ***Proceedings of the International*** *Conference on Computing in High Energy Physics (CHEP'01)*, *Beijing, China, 3–7 Sept. 2001*, p. 109.

[23] N.Klopov, E.Novodvorsky, S.Olechko, and Yu.Ryabov, in *Proceedings of ATLAS TRT Workshop, Istanbul*, *Turkey, 3–9 June 2001.* 1http://edmsoraweb.cern.ch:8001/cedar/doc.download?document_id=311734&version=1&filename=end_cap_prod_db_status.doc>.