Design and Development Issues of Retrospective Database in Complex Technological Control Systems

Dmytry Kuklenko
National Technical University
“Kharkiv Polytechnic Institute”
Frunze str. 21
Ukraine-61002 Kharkiv
dkuk@kpi.kharkov.ua

Abstract: One of the aspects of the reengineering problem of legacy Information Management Systems for complex Technological Process, related with working out of new program components, destined for storing and processing the retrospective information concerning the functioning of Technological Process, was considered. As an object for re-engineering processing the Computer-Aided Information System (CAMS) of one of the plants of Romny (Ukraine) Gas Compress Station. A new CAMS program architecture, the model of new program component, which was named as Retrospective Database, and main aspects of its software implementation are considered in this paper.

1. Introduction

Nowadays, the class of Information Management Systems (IMS) for complex Technological Process (TP) is developing impetuously. The application of large complex systems like robotized manufactures; movement management systems and other subcategories of them are now considered to be custom things. At the same time, the development and implementation of such system requires substantial material expenditures. Therefore, it can be much more efficient to adapt the existing (legacy) IS for the conditions and requirements, or to accomplish an IS re-engineering.
During last two years the Reengineering-Research Group (RRG) of the National Technical University “Kharkiv Polytechnic University” (NTU “KPI”) is dealing with the reengineering issues related to some legacy IMS operating in Ukrainian main gas-pipe line system [Tk99],[Tk00].

2. The short description of “Romny” GCS – predestination, structure, existing IS and its drawbacks

One of such complex technological objects is “Romny” Gas Compress Station (GCS). This station is situated at the northeast of Ukraine and intended for gas pressure maintenance in arterial gas-pipeline, which conveys natural gas from Russia’s fields to

Western Europe. The UML-diagram, which represents the structure of GCS[DOO99], is shown on Fig. 1.

"Romny" GCS includes several plants, which represent the set of gas -pumping aggregates. In this paper the main plant of "Romny" GCS with internal name "KS32P" was considered.

Each plant provides the gas pump ing-over in one specific branch of pipeline. These gas-pumping aggregates are supplied by a large number of special technical subsystems, and all these technical components can be represented as technological processes. Farther, the plant block also contains the Technological Process Computer-Aided Management System (TP CAMS). This TP CAMS includes a number of subsystems, which are connected with corresponding TP. We can emphasize about 20 subsystems in KS32P plant (for e.g., Common Station Equipment (CSE) CAMS, Gas Automatic Air Cooling (AAC) CAMS and others). Each technological process, also, is characterized by ample quantity of technological parameters (number of them can be estimated in tens or hundreds), values of which are important for estimation of plant functioning quality.

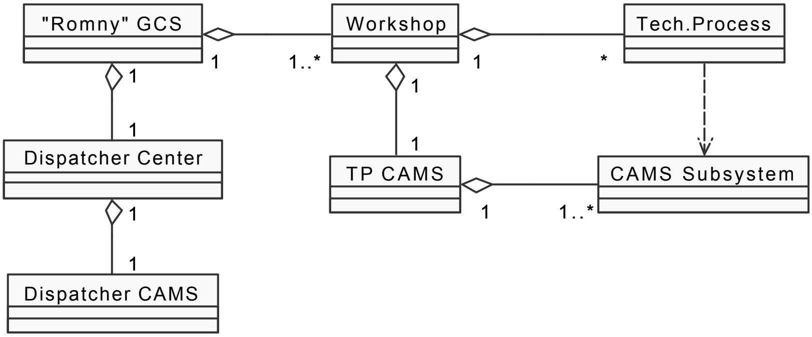

Fig. 1 – The structure of "Romny" GCS. UML class diagram

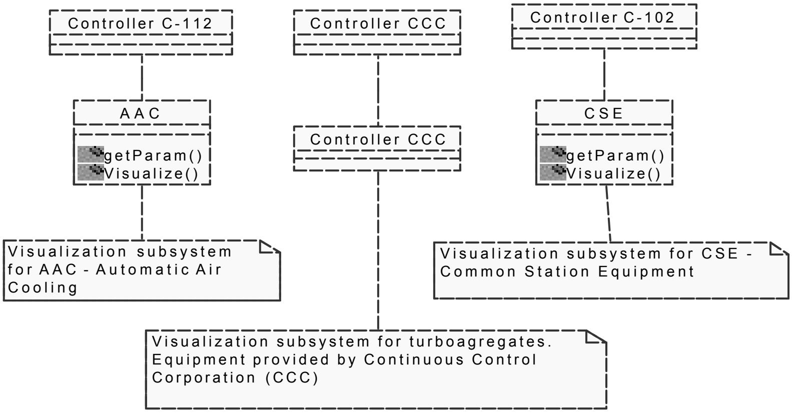
Controller C-112
Controller CCC
Controller C-102
AAC
getParam()
Visualize()
Controller CCC
CSE
getParam()
Visualize()
Visualization subsystem for AAC - Automatic Air Cooling
Visualization subsystem for CSE - Common Station Equipment
Visualization subsystem for turboagregates. Equipment provided by Continuous Control Corporation (CCC)

Fig.2 – The CAMS of KS32P plant. UML class diagram

Considering the CAMS of KS32P Plant (its UML class diagram is shown on Fig. 2), we can note the following drawbacks of it [DOO99]:

The KS32P plant CAMS represents a set of incoherent components. These components do not provide any unified interface for system operators and do not have any common data.

The components of CAMS being described provide only the data visualization, and the TP parameters values are only reflected on the monitor, but they are not stored in any depot, so there is no possibility for analysis of TP progress.

Because of incoherence of these components and the lack of data storage, it's impossible to make some aggregate estimation of plant state.

The information about current state of plant cannot be granted to any remote CAMS (for e.g., to Dispatcher CAMS).

3. An installing of retrospective database components as an approach of KS32P plant CAMS reengineering

One of the possible ways of this IS improvement is an implantation of some new component, which will serve as data storage for existing CAMS subsystems. This component can be named "Retrospective Database". This component is intended for the following:

1) Collection and accumulation of the active information about the GCS technological process. It also gives the possibility of further graphical representation and analysis of these data.
2) Automatic aggregation of parameter values after the expiration of some pre-defined time periods. This function is realized by the calculation of aggregate average values of technological process parameters per every hour, day, month also. It lets the data analysis procedures to be more simple and allows us to consider the RDB as an information system , which was developed with the use of data warehouse technology.
3) RDB subsystem can be applied for the researches of cause of failures. Some situations, which can occur during the GCS functioning, can be considered as failures. For e.g., it can be the sudden gas pressure fall in gas pipeline (it can mean that the pipeline is broken), quick rise of temperature in some gear pulley (it can mean that there is a fire) and so on. When RDB subsystem records the failure situation, it stores all the parameter values from some pre-defined time period (10 minutes, for e.g.) before and after the failure. These data can be used afterwards for the detailed analysis of failure causes and of technological process evolution after the failure. The result of this analysis can prevent the rise of similar failure in future.
4) Current version of RDB subsystem was developed for CSE (Common Station Equipment) of K32P plant of Romny GKS. But, because of RDB design features, it can be integrated in the existing IS of other GKS plants and blocks. In this way we can provide the re-engineering of legacy workable software by adding the new subsystems for storing data.

Depending on the RDB purposes, we can make a conclusion, that RDB should have a mixed operational-analytical structure, so, it should have both the features of OLTP and OLAP systems architecture (for routine and archive data, conformably).

4. A new system architecture and its description

The problems, which relate to the implementation of OLAP system, are well-known (see, for e.g. [De97], [Ed97],[Lv97]). The following ones can be distinguished among them:

Different external sources data formats matching;

The problem of one time scale fixing;

The processing of extra-large data size;

Unstandarized queries of analytics and experts, etc.

The attempts of overcoming these problems leaded to appearance of the Data Warehouse approach and Information Warehouse conception [Mu98]. Practically, the use of one of these approaches by choosing the specific IS implementation manner inevitably makes the developers solve the problems of modifications or integration of existing relational (in most cases of DB).

One of the possible architecture of a system, which serves for decision making support [Lv97], is shown on Fig. 3.

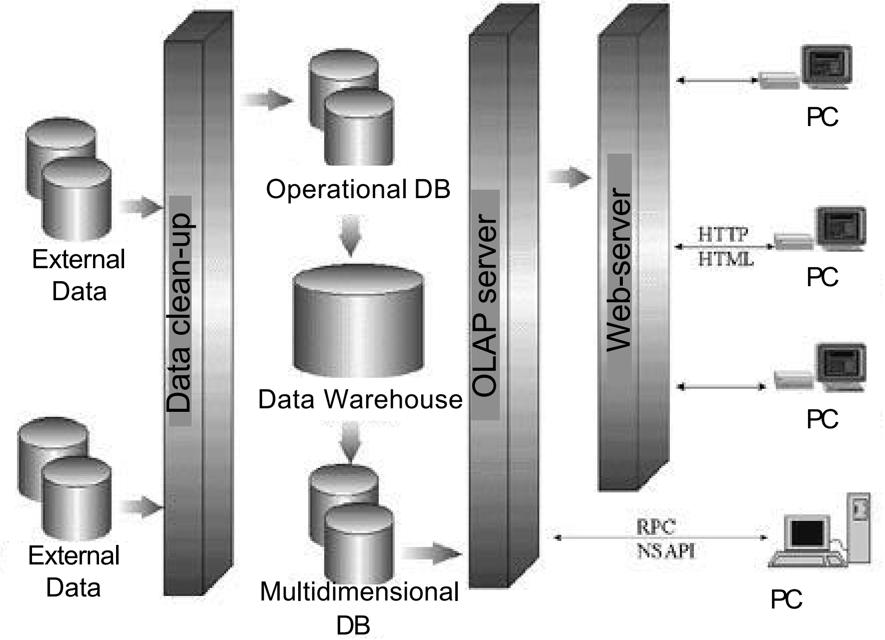

Fig.3 – Possible architecture of decision making support system

This type of architecture can be used in new KS32P CAMS structure. There can be, in example, correspondence between real CAMS components and structure elements mentioned above:

External Data – the data, which are getting from system controllers

Data Clean-Up – procedures of data processing and parameters values and system events allocation

Operational Database – routine data (current parameters values and system events)

Data Warehouse / Multidimensional Database / OLAP Server – archive data (aggregated parameters values per hours and days and initial parameters values, which are important for failure analysis)

Local Users – GCS operators, engineers and managers, who need the information about current situation on GCS

Remote Users – GCS dispatchers and managers, as well as higher organs experts, who need the station archive information.

According to considerations mentioned above, the new system architecture with RDB looks as shown on Fig. 4.

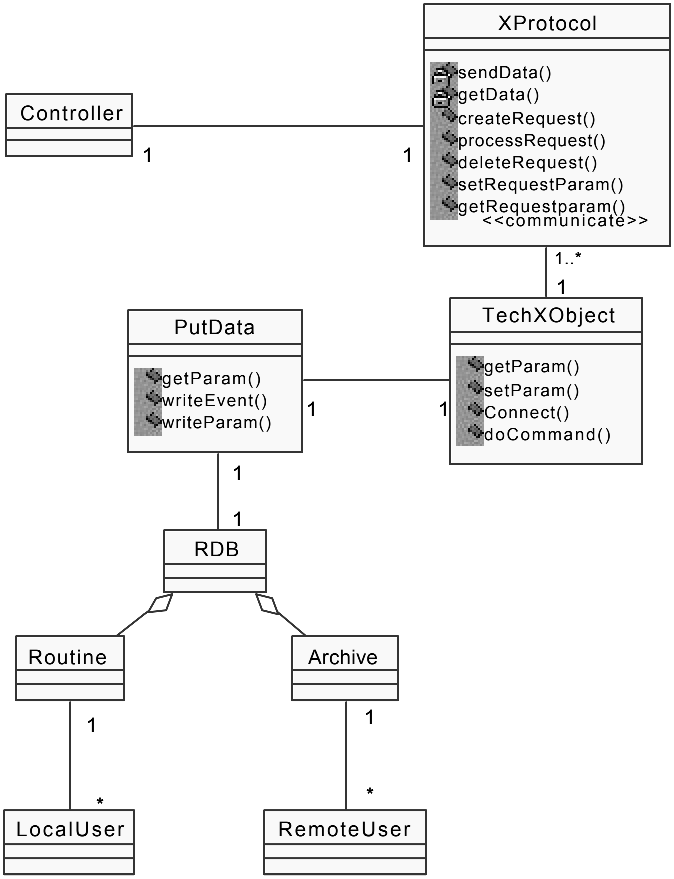

Fig.4 – The fragment of a new system architecture with RDB

This architecture includes several new components, except RDB. Let us make a short description of them:

1) *XProtocol* - WinNT service, which assures the interaction between whole system and PC port. The class CXProtocol is created for each PC port, which is used for data intake in the system. Also this component allows not only read the data from PC port, but also send the data through PC port (for example, it can be used for determining of the settings values).

2) *TechXObject* - WinNT service, which provides the interface for interaction between Xprotocol objects and final applications. This component allows to resolve the problem with data synchronization and representation (data conversion from binary representation into decimal).

3) *PutData* - MS Windows NT application, which reads the data from TechXObject component and writes them into Retrospective Database.

5. The RDB Routine section

Let us consider the structure of Routine RDB section. RDB subsystem structure is object-oriented. We can mark out the following classes:

Device class – realizes the physical device entity of the plant (for e.g., ventilator, pump)

Parameter class – realizes the device parameter, values of which have to be stored (for e.g. – the temperature at the outlet tap, the gas pressure at tap #20)

Event class – realizes the system event. System event is a binary parameter, which indicates the presence or absence of some system state (for e.g., “The tap #15 is open”, “The temperature of tap #17 is high”)

These three classes realize the system metadata (the data about the system structure), so it allows to suppose this RDB structure as universal and adjustable for any similar technical object with similar structure.

TimeMoment class – realizes the entity of real time.

ParameterArchive and EventArchive – realize the storage of parameters values and events occurred.

The UML Class diagram of RDB component Routine section is shown on Fig. 5.

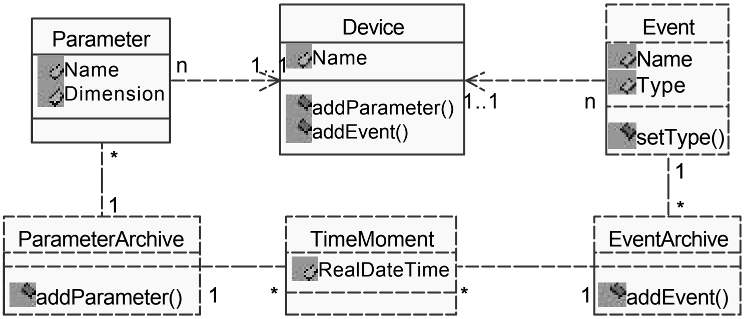

Fig.5 - The UML Class diagram of RDB component Routine section

The corresponding IDEF1X diagram is shown on Fig.6.

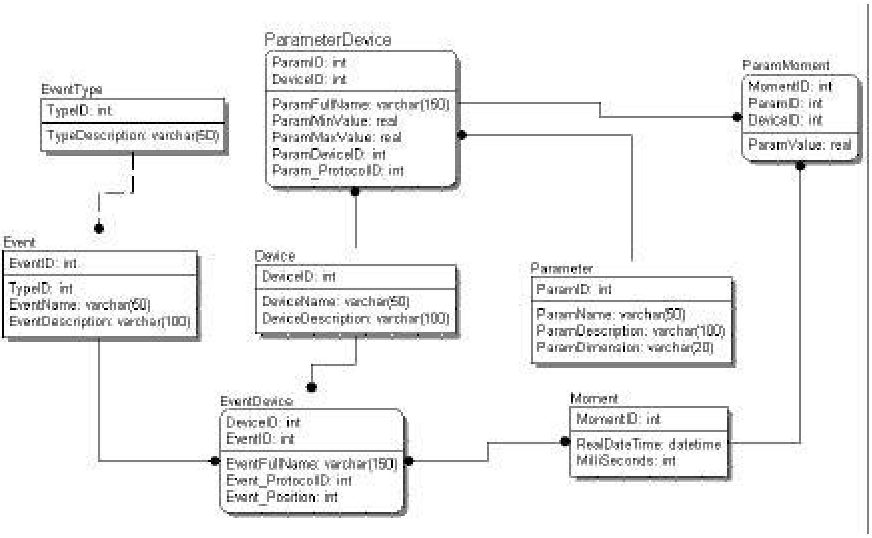

Fig. 6 – The IDEF1X diagram of RDB Routine section

6. The RDB Archive section and the description of archiving process

The RDB Archive section provides the parameters average values storing and also the values from failure ranges. The IDEF1X diagram of RDB archive section is shown on Fig. 7 and the schematic algorithm of data archivation is shown on Fig. 8.

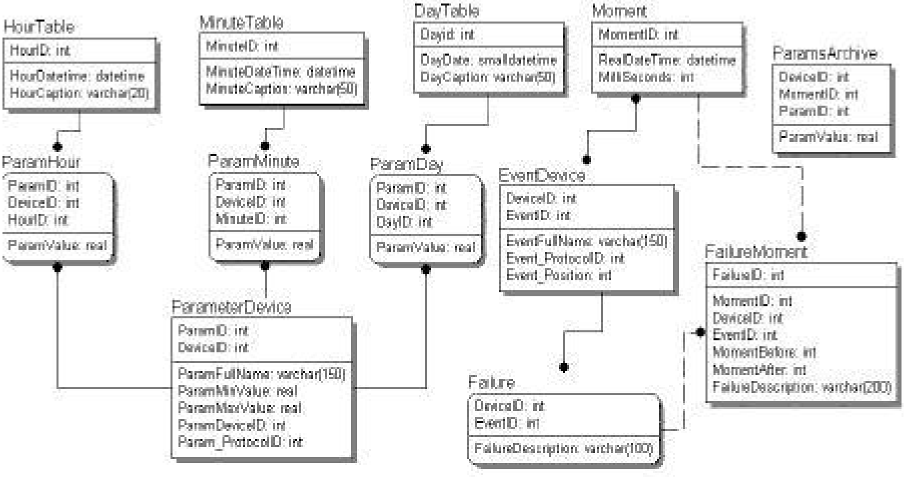

Fig. 7 - The IDEF1X diagram of RDB Archive section

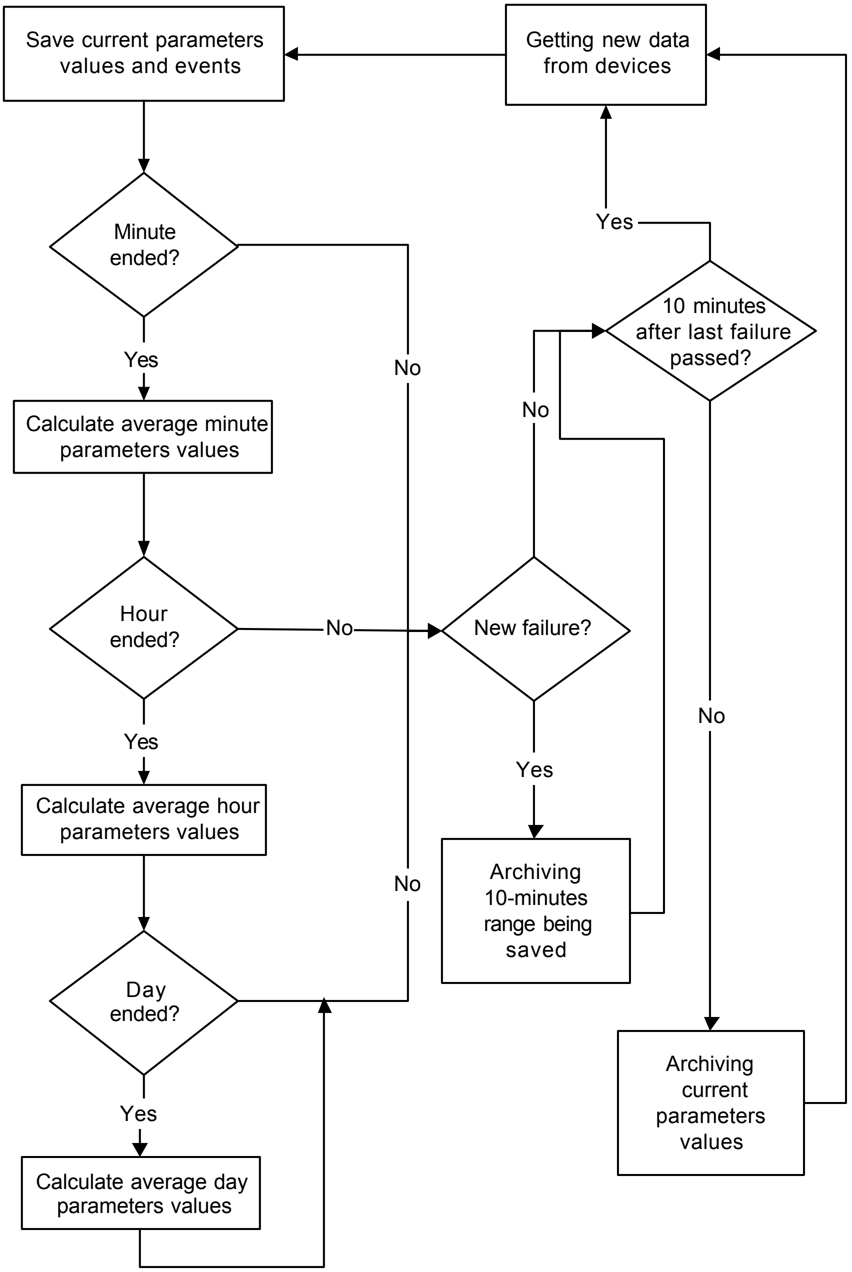
Save current parameters values and events
Getting new data from devices
Minute ended?
Yes
No
Calculate average minute parameters values
Hour ended?
No
Yes
Calculate average hour parameters values
Day ended?
Yes
No
Calculate average day parameters values
10 minutes after last failure passed?
Yes
No
New failure?
No
Yes
Archiving 10-minutes range being saved
Archiving current parameters values

Fig. 8 – The schematic algorithm of data archiving

7. A new CAMS program architecture and some aspects of RDB implementation

The new program architecture for "Romny" GCS KS32P plant is shown on Fig.9.
The RDB version which was installed during the process of "Romny" GCS re-engineering was created with the use of MS SQL Server 97 DBMS. The correspondence between the actions from data archivation algorithm and used program tools for their implementation is shown in the table 1.

Table 1. The correspondence between algorithm actions and program tools

Algorithm action	Implementation	Comment
Saving parameters and events	SQL Server Stored Procedure	
Checking, if minute (hour, day) ended	SQL Server Trigger	If ended, trigger executes the corresponding stored procedure for data aggregations
Calculation of average parameters values	SQL Server Stored Procedure	
Checking, if new failure is occurred	SQL Server Stored Procedure	If occurred, procedure returns TRUE value of output parameter into PutData
Archiving a 10-minute range being saved	SQL Server Stored Procedure	Is executed from PutData thread being run from PutData, when failure occurs
Checking, if 10 minutes after last failure passed	PutData component	Using the data from RDB
Archivation of current parameters values	SQL Server Stored Procedure	If last failure happened less then 10 minutes before
Getting new data from devices	PutData component	Using the TechXObject service

8. Conclusions and perspectives

Naturally, after RDB collects and archives information for comparatively big time period, the main problem will be to realize the analysis of these data in order to reveal a hidden appropriateness in archive data (for e.g. what kind of changes in data values can cause a system failure, and what can be referred to changes in external conditions, etc.) and to forecast the future failures on the base of data being archived. The neural networks methods [Ko00], [Fa94], [Ha99] and genetic algorithms [Ar98] are now planned to use in this part of work. And, of course, the RBD subsystem should become the essential part of the *System Reengineering Information Environment Framework*, which is now elaborating by RRG NTU "KPI" for large legacy IMS.

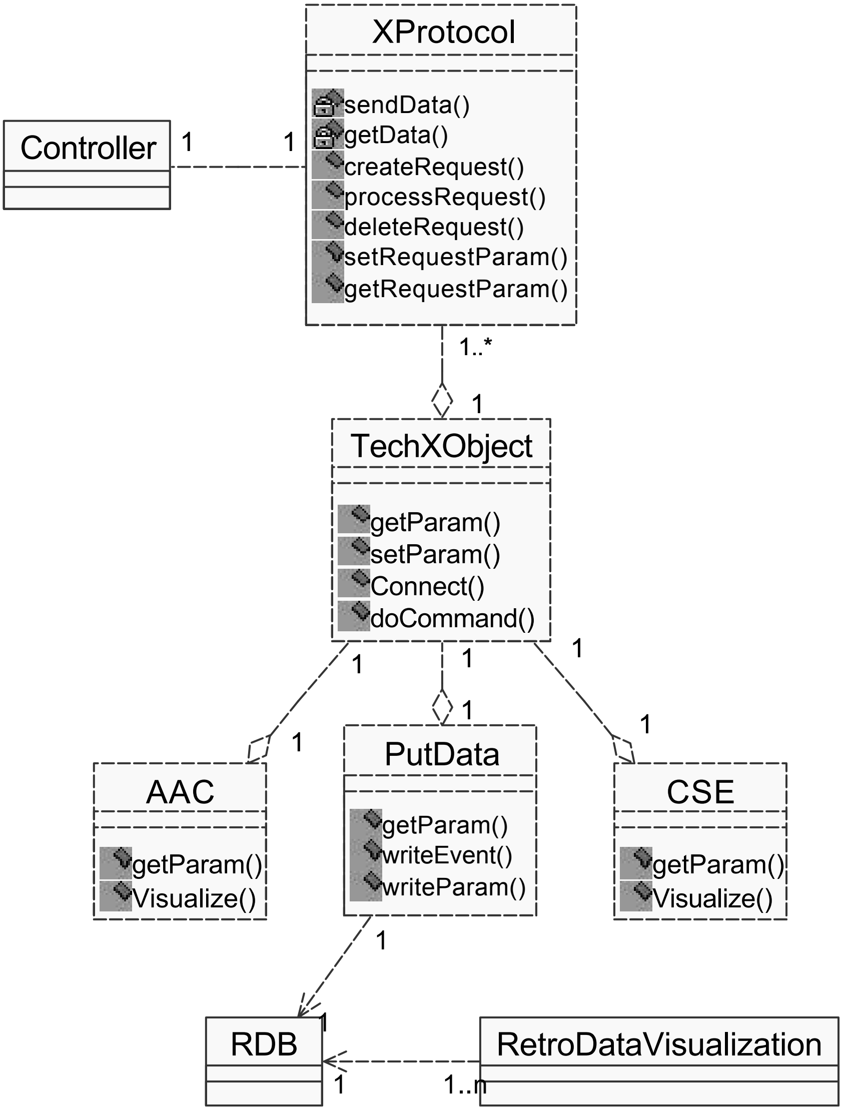

Fig. 9 - The new program architecture for “Romny“ GCS KS32P Plant

9. Acknowledgements

I would like to thank Dr. N.Tkachuk for his valuable aid in my work. Also I’d like to thank the technical director Mr. S. Ovasapov and the chief engineer Mr. A.

Kuzmin of the design firm “PromAvtomatika” (Kharkiv, Ukraine) for their stimulating discussions about some problems, presented in this report.

Bibliography

[Tk99] Tkachuk N.V., Habboush A.Kh., Popov S.V., Dudko P.G. Object - oriented structural model and program package for distributed control system of air cooling facility on gas-compressor station. // System analysis, management and information technologies: Bulletin of KhSPU. The anthology of scientific papers. Issue #73. – Kharkiv, KhSPU. –1999. Pages 3-7.

[Tk00] Tkachuk M., Kaschek R., Popov S., Habboush A. Kh. A Gas - Compressor Station Case Study in Software Re-engineering // ReTIS2000: 6-th International Conference on Re-Technologies for Information Systems (Preparing to E-Business) . v Oesterreichische Computer Geselschaft. Band 132. - 2000. v p. p. 142-153.

[DOO99] Dolgopolov S., Orlov A., Ovasapov S. The integrated system of computer-aided control of gas compressor station common equipment // System analysis, management and information technologies: Bulletin of KhSPU. The anthology of scientific papers. Issue #73. – Kharkiv, KhSPU. –1999. Pages 33-37 (in Russian)

[De97] Barry Devlin “Data Warehouse: from Architecture to Implementation”, Addison - Wesley, 1997

[Ed97] Herbert A. Edelstein, Ramon Barquin “Building, Using, and Managing the Data Warehouse”, Prentice Hall / 1997

[Mu98] URL : http://money.ct.ru/user/st1.html- A.Musaev “Intellectual data analysis – Klondike or Babylon?” (in Russian)

[Lv97] URL : http://www.idguniverse.ru/dbms/1997/03/30.htm - V.Lvov. “The working out of decision support management systems based on data warehouse technology”. (in Russian)

[Ar98] URL : http://www.bizcom.ru/rus/bt/1998/nr3/14.htm S.Archipenkov “From the data processing to its analysis” (in Russian)

[Ko00] V.V.Korneev, A.F.Garev, S.V.Vasyutin, V.V.Reih “ Databases. Intellectual data processing”. Moscow, “Knowledge”, 2000 (in Russian)

[Fa94] Laurene Fausett “Fundamentals of neural Networks: Architectures, Algorithms and Applications” – Prentice Hall / 1994.

[Ha99] Simon Haykin “Neural Networks” – Prentice Hall, 1999.