APPLICATION OF PyCDB FOR K-500 BEAM TRANSFER LINE*

Pavel Cheblakov#, Sergey Karnaev,
Oidin Khudayberdieva, BINP SB RAS, Novosibirsk, Russia

Abstract

The new injection complex for VEPP-4 and VEPP-2000 e-p colliders is under construction at Budker Institute, Novosibirsk, Russia. The double-direction bipolar transfer line K-500 of 130 and 220 meters length respectively will provide the beam transportation from the injection complex to the colliders with a frequency of 1 Hz. The designed number of particles in the transferred beam is 2*10^10 of electrons or positrons, the energy is 500 MeV. K-500 has dozens of types of magnets, power supplies and electronic devices. It is rather complicated task to store and manage information about such a number of types and instances of entities, especially to handle relations between them. This knowledge is critical for configuration of all aspects of control system. Therefore we have chosen PyCDB to handle this information and automate configuration data extraction for different purposes starting with reports and diagrams and ending with high-level applications and EPICS IOCs' configuration. This paper considers concepts of this approach and shows the PyCDB database structure designed for K-500 transfer line. An automatic configuration of IOCs is described as integration with EPICS.

INTRODUCTION

Every large scientific facility is a complicated construction with a huge amount of hardware, electronic equipment and software. Hardware is controlled by built-in electronics in some cases, it requiring dedicated or universal standalone controllers in other cases. Anyway it results in using of a wide spectrum of electronics of different types from different vendors. Moreover it can rapidly increase the amount of physical protocols types (field busses), interconnections, network equipment, CPU and OS platforms and etc. Consequently, software should provide support for diversity of electronics and thereby hide its complication from user.

The largest facilities (e.g. ITER [1]) include hundreds of local systems and dozens of subsystems. These local systems consist of a set of controllers, interfacing the actuators and sensors, and connected together via network switches to the plant operation network. The total amount of computers can reach one thousand, the total number of signals (wires) and the total number of process variables is estimated at 100.000 and 1.000.000 accordingly.

* This work has been supported by Russian Science Foundation (project N 14-50-00080).
# P.B.Cheblakov@inp.nsk.su

PROBLEMS

To maintain so large and diverse system it is necessary to collect and store all relevant information: not only lists of systems, subsystems, network equipment, controllers, instrumentation and control devices but relationships between them as well. We need to know which controller is responsible for corresponding power supply or specialized gauge. We need to know how that controller is connected to corresponding computer and which switch is used. Also we need to know what types of cables are used for interconnection.

Strictly speaking we can continue this list further and further adding more and more information about our facility. This information is vital for broad range of specialists: from storekeepers, designers and electricians to software developers, operators and physicists. Having this data we can create a list of equipment of particular type for record keeping purposes. Or we can simply generate actual cable schedule. Furthermore we can get configuration data for software involved in control.

This information is mostly distributed along many spreadsheets and databases which are weakly connected. This fact is likely to cause data duplication and denormalization so we can get data inconsistency induced by human factor. As a result it is very difficult to change and analyse distributed data both for human and computer.

There are different approaches to store and handle this information. First of all we divide the information into two parts: the first one is used for software configuration and the second one includes everything else. Handling the former part of data is more formalized and there are some practices to operate with it. For example the paper [2] distinguishes centralized and decentralized approaches covering almost all cases and proposes hybrid technique. Even though it turns to a laborious work for every facility [3, 4, 5].

As regards the latter part of the data it is so unstructured and non-formalized that it becomes a great challenge to build up a data-storing system from scratch or even deploy any specialized software system [6].

KNOWLEDGE BASE

To deal with heterogeneity and complexity of data we propose to use a centralized storage which is able to store and handle with ease not only data and its properties but relationships as well. Storing and retrieving relationships is a key aspect for successful functioning of many systems: from equipment procurement and registering to cable scheduling, from GIS-data to software configuration

ISBN 978-3-95450-148-9

system. Let us use a name *knowledge base* for our storage instead of extensively used term *database* to distinguish the former from classic relational database [7]. We need to store not just tables with numbers and strings but pointers to other objects which in turn have additional pointers.

Therefore our knowledge base represents facts about scientific facility in a structured form that suitable for software processing and analysis. An object model often called *ontology* can be used for representation of knowledge base [8]. Ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain.

GRAPH DATABASE

Closely connected conception to knowledge base is a *semantic network*. It is a network which represents semantic relations between concepts. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges [9]. Applying ideas of semantic network to our problem domain - scientific facility - we can create *facility informational model*. Since semantic network is a graph we assume that using a graph database is a good choice.

Graph databases stay far away from relational databases and they differ markedly from most NOSQL databases. They have some drawbacks which are mostly not relevant to our problem. But advantages make graph databases a preferable solution. One compelling reason for choosing a graph database is the sheer performance increase when dealing with connected data. Graphs allow structure and schema to emerge in tandem with our growing understanding of the problem space, rather than being imposed upfront, when we know least about the real shape and intricacies of the data. Graphs are naturally additive, meaning we can add new kinds of relationships, new nodes, new labels, and new subgraphs to an existing structure without disturbing existing queries and application functionality [10].

K-500 TRANSFER LINE

K-500 is a transfer line for electrons and positrons which is under construction in Budker Institute of Nuclear Physics, Novosibirsk, Russia. The source of particles is Injection Complex (VEPP-5) and it will supply both BINP colliders: VEPP-4 and VEPP-2000. Right now we are working on the part from Injection Complex to VEPP-4 and it contains 5 DC power supplies and 15 pulsed power supplies of 5 types in all. 24 electronic devices of 8 types are used to control power supplies and measure their parameters.

PyCDB [2] was chosen to operate with graph data. This is a specially designed graph database engine to operate over facility informational model. The data model remains almost the same as developed for the booster of the NSLS-II [11] while we tried to make it more flexible (see Fig. 2).

The special tool was developed for editing and viewing graphs. It is a graphical web-application which allows to observe and manipulate with data in a graph stored in PyCDB. It is shown on the Fig. 1. The main window has three parts. The left side contains a data model for corresponding facility, the right side (largest one) is a graph itself and the upper part allows us to edit properties and "look and feel" of selected entity or relationship.

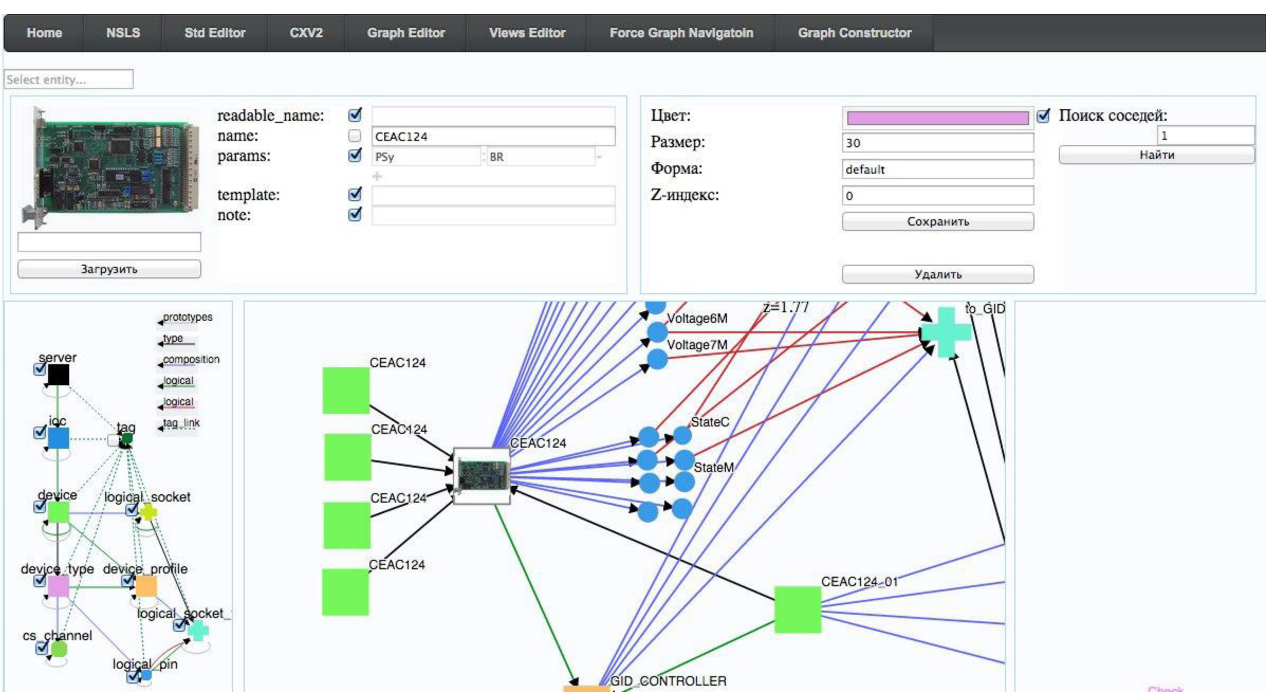

Figure 1: The Graph Editor's main window.

ISBN 978-3-95450-148-9

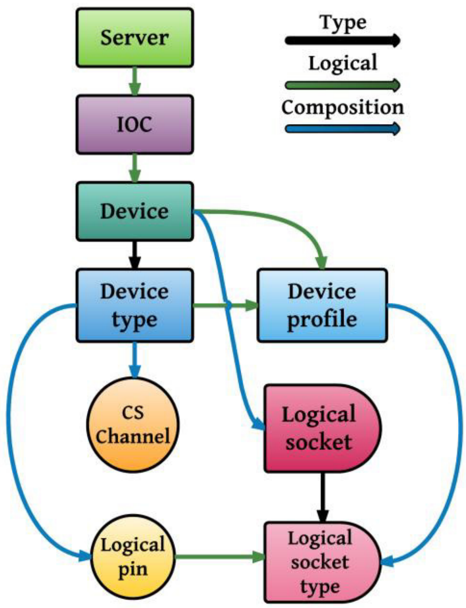

Figure 2: The conceptual data model for K-500's Power Supply System.

Control Systems Integration

One of the obvious and desired possibilities for applying facility informational model is to automatically configure EPICS IOC Database [12]. Special Python applications are able to fetch data from the database and generate IOC's start-up scripts. These scripts are loaded into IOC and define a part of distributed EPICS database.

CONCLUSION

Using technologies initially developed in a scope of artificial intelligence, particularly for expert systems and knowledge systems, may meet requirements of modern scientific facilities. Complexity and scale of recent ones makes handling all relevant information to be a quite complicated task. Appearing and fast growing of modern technologies like graph databases gives a new life for old ideas. Applying a graph database for creation and handling facility information model for K-500 transfer line is in progress. A high necessity of such model is shown, a graphical editor for a model was developed and made some steps towards control system integration.

REFERENCES

[1] A.Wallander et al., "News From ITER Controls – a Status Report", proceedings of ICALEPCS2011, Grenoble, France (2011).

[2] A.Makeev et al., "Centralized Software and Hardware Configuration Tool for Large and Small Experimental Physics Facilities", proceedings of ICALEPCS2013, San Francisco, CA, USA (2013).

[3] R. Bakker et al., "Significance of a Comprehensive Relational Database System for Modern Accelerator Controls", ICALEPCS97, Beijing, October 1997, p. 239.

[4] Z. Zaharieva et al., "Database Foundation for the Configuration Management of the CERN Accelerator Controls Systems", proceedings of ICALEPCS2011, Grenoble, France (2011).

[5] D. A. Dohan et al., "Integrated Relational Modeling of Software, Hardware, and Cable Databases at the APS", proceedings of ICALEPCS2003, Gyeongju, Korea (2003).

[6] D. Stepanov et al., "Integrated Approach to the Development of the ITER Control System Configuration Data", proceedings of ICALEPCS2011, Grenoble, France (2011).

[7] Krishna, S. Introduction to Database and Knowledge-base Systems. Singapore: World Scientific Publishing, 1992. ISBN 981-02-0619-4.

[8] Oberle, D., Guarino, N., & Staab, S. "What is an ontology?". In: "Handbook on Ontologies". Springer, 2nd edition, 2009.

[9] John F. Sowa (1987). "Semantic Networks". In Stuart C Shapiro. Encyclopedia of Artificial Intelligence.

[10] Ian Robinson, Jim Webber, and Emil Eifrem, "Graph Databases". 2nd edition. 2015.

[11] P.Cheblakov et al., "Configuration System of the NSLS-II Booster Control System Electronics", proceedings of ICALEPCS2013, San Francisco, CA, USA (2013).

[12] R.Keitel, "Generating EPICS IOC Data Bases from a Relational Data Base – A Different Approach", proceedings of ICALEPCS'01, San Jose, CA, USA (2001).

ISBN 978-3-95450-148-9