The unified software package Sonix+. Analysis of development experience

Frank Laboratory of Neutron Physics, Joint Institute for Nuclear Research, 141980 Dubna, Moscow Region, Russia.
E-mails: akirilov@flnp.jinr.ru, akirilov@jinr.ru

The software package Sonix+ (Sonix) [1] developed as a unified control software for neutron instruments. It has been in operation since 1995. Currently it is used at instruments of the IBR-2 (FLNP, JINR) and at some other centers of the Russian Federation (totally about 20 installations). Though the main ideas of the Sonix+ are largely similar to the decisions taken at other centers of the NOBLEUGS community, the consideration of specific needs and traditions prevailing at the FLNP, had substantial influence at the structure of the complex, and some implementation decisions. This paper, for example, the choice for the operating system of instrument control computer, implementation details of so-called “database” – parameter storage with fast access for inter-module communications, a graphical user interface, choice of the spectra recording format, etc. During the long period of exploitation the complex was developed, it has been added new components and has been changed to the new requirements. This led, in particular, to the adjustment of a number of initial decisions, and to a certain eclecticism. Some implemented features have unclaimed, and others, on the contrary, have required further development.

Few words about history

The Sonix+ project was started in 1987. It was maintained at the Sonix+ Laboratory of High Resolution (SRB) at beam line 7s of the pulsed reactor IBR-2 of the IJBR in March 1990. A UNIX based computer running OS/2 was used for the instrument control. The GUI was built using the Microsoft Windows interface and a number of libraries including the Microsoft Control Panel. The software was later adapted for several other instruments and was named the Sonix (SOrian software for Neutron instruments on the K111). The adaptation was made using COM because it was a standard way of managing the software as an universal one. For example, in order to provide an ability to organize hierarchical structures of execution modules, the custom made script was rather poor, there was no visualization of spectra for the Varman user interface of the corresponding Varman server.

In the early 2000s it was decided to replace the expensive control VME computers with cheaper PCs, and the Ultra-idea Delphi operating system with the Windows NT. This stimulus re-elve to create our control software due to obtained exploration experience in the recent trends. The new version is called Sonix+. Currently the Sonix + is the set of modules, unified by the common philosophy, which allows to organize control system for arbitrary instrument rather simply. Sonix+ GUI also has undergone a long evolution from separate window for each controllers to a single common window. There is the Command line propose to enter commands as a Python string using any of standard shell (mostly PythonWin client).

Main features of Sonix+:

The Sonix+ is a unified, universal modular set of modules. It has many features both in structural principles and implementation ideas more or less similar to those of modern control software [3]. A main difference is the Sonix + is mostly local system. Computation power of modern computer usually is sufficient to do the whole job, this decision simplifies the system implementation. For those applications that require synchronization with external computers as called “channel” (channel command) module is provided. In our practice this is very seldom. For remote user access several possibilities are available: the WebFront interface, the VNC, the WebSonic service [4].

The Microsoft Windows is used as an operating system on control computers. This choice was made on the urgent wish of our users, many of whom preferred the Ultra-idea Delphi operating system very urgently. Another reason for this choice was the advantage in the software development systems, compared with the Linux at that time. Unfortunately, there is a separate data format for each IBR-2 instrument up to now. So, the Novus standard [5] for storing measurement results were not chosen because our users do not interested in it. In this situation we decided our own format as control of the node is not for all instruments. It consists two files, one of which contains the actual spectra and the second contains the measurement history (log file). This image contains all measurement parameters registered in the Sonix+ Final data are transferred to format specific to each experiment.

We have proposed and implemented a universal approach to GUI [6]. This means that the same set of clients are sufficient to satisfy requirements of an arbitrary instrument without change.

Sonix+ Structure

The Sonix+ structure is presented in the Figure 1. In a modular system every component module is responsible for some device or function. In practice, a set of modules for each instrument has a non-linear structure. Some modules within the same device – they are at a lower level of the hierarchy – they can be modified by the man of others. We have chosen these components and tried to create their new version as an universal one. We called these modules servers. If it takes to port Sonix+ to another instrument, servers can be used without radical changes.

Of course, for communication of servers with other modules specialized protocols are needed. Actually the following protocols have been excluded:

- universal protocol for device control and querying:
- OAI protocol:
- motor protocol:
- remote inquiry and control protocol via sockets.

All of these were proved by the long practice with the exception of motor server. It will be eliminated in the nearest future.

Variable manager (Varman)

During the experiments, all modules communicate through the special storage. It is called Varman database. This storage provides very big volume of data. At any moment the contents of this Varman database completely represents the current state of the measurement process at the instrument with the exception of generally.

The Sonix+ structure was taken from I/ODI specifications. The Windows was made by V. Vrubel [7]. Initially all communications were made through simple variables like “motor position”, etc. Now operations mostly deal with “structures” of data. This trajectory is consistent with visual concept of the universal Varman+ GUI. Efforts have been written on the base of PyQt and the graphics library matplotlib.

The Sonix+ structure

The ICE (In-Beam Control and Experiment) of the Sonix+ project success is proved by its practice. At the same time, as would be expected, some problems have been encountered at the structure of the complex, and some implementation decisions have required further development.

Spectra visualization

Spectra visualization includes a set of widgets for visualization of mono detector, 1D PSD and 2D PSD data. Spectra can be read from files (including zipped files) or from the DAQ controllers directly. The resizable library panel is used to display the graph. Besides curve drawing the panel implements typical operations like scaling, shifting, etc. Additional operations implemented in the widgets are automatic/manual definition of limits in the display window, some others concerning dimensionality of the data.

Instrument tuning

The tuning program ICE is developed for tuning of Varman servers. The ICE program is implemented as an add on the control complex Sonix+. The problem was to adjust the instrument before the measurement. Program have been written using the PyQt and the graphics library matplotlib.

User interface

There are generally three kinds of Sonix+ interfaces: command line, GUI and web interface. Command line interface provides the most powerful commands as a Python script for any experiment. The most powerful PythonWin client. Sonix+ GUI also has undergone a long evolution from separate window for each controllers to a single common window. There are universal module structures for each module. The universal GUI is available for all modules. This GUI is able to control any instrument – each sub-window is dedicated to one of the main users’ needs. There are three main windows to select: current state of the instrument, the measurement configuration and the measurement process. The fourth window is to control the measurement process. The fourth programs (windows, widgets) are generally sufficient to conduct an experiment. There are additional programs as well (the Load control panel, the Configuration editor and some others).

Specialized GUIs are created by special request, for instance, for instrument tuning. Some specialized interfaces were created for instruments at other centers, if these users are known to be far from modern computers.

Script utilization

In current version the language is used as a script language in the Sonix+ and all modules features and widgets of Python are available to control experiments and for preliminary data processing.

This allows one to reasonably separate the specificity of measurements instruments into the most flexible and easy-to-change component. This fact is very useful from the viewpoint of utilization and not tokenizing the transfer of Sonix+ to new instruments.

It is also important that using the Python as a script language is a wide door for the further development of virtual programs. Unfortunately, during these years there were very few ideas for development of models.

The Sonix+ script is the pure Python code. To control the script interpretation a special module is developed. At other places of the system, it allows to extend the existing models.

The Sonix+ project is widely used. At the same time, as would be expected, some of the previous decisions have to be revised, in accordance with emerging needs and opportunities.

Akknowledgements

For a long period of development and use of Sonix+ many people in the FLNP and almost contributed to the project with their work, ideas and interests. The author is deeply grateful to all of them.

References

The universal GUI (in the center) and the component set

The ICE tuning program illustrations

The YMD instrument script example

Future plans

The main direction of our future efforts is organization of centralized repository for measurement data with corresponding services.

Conclusion

Overall the Sonix+ project success is proved by its practice. At the same time, as would be expected, some of the previous decisions have to be revised, in accordance with emerging needs and opportunities.

The unified software package Sonix+. Development and use experience