

## Abstract

At FLNP (JINR, Russia) there is a pulse reactor IBR-2 with multiple spectrometers included thereto, controlled with the help of the instrumental software complex Sonix+.<sup>1</sup>

The Sonix+ complex includes controlling modules and programs with a universal user interface.

Although these programs perform the whole set of necessary functions (review of the measurement, script control, visualization and etc.) , they cannot be used for the development of the programs for adjustment or of new specialized programs. At the same time FLNP's need of new programs is considerable.

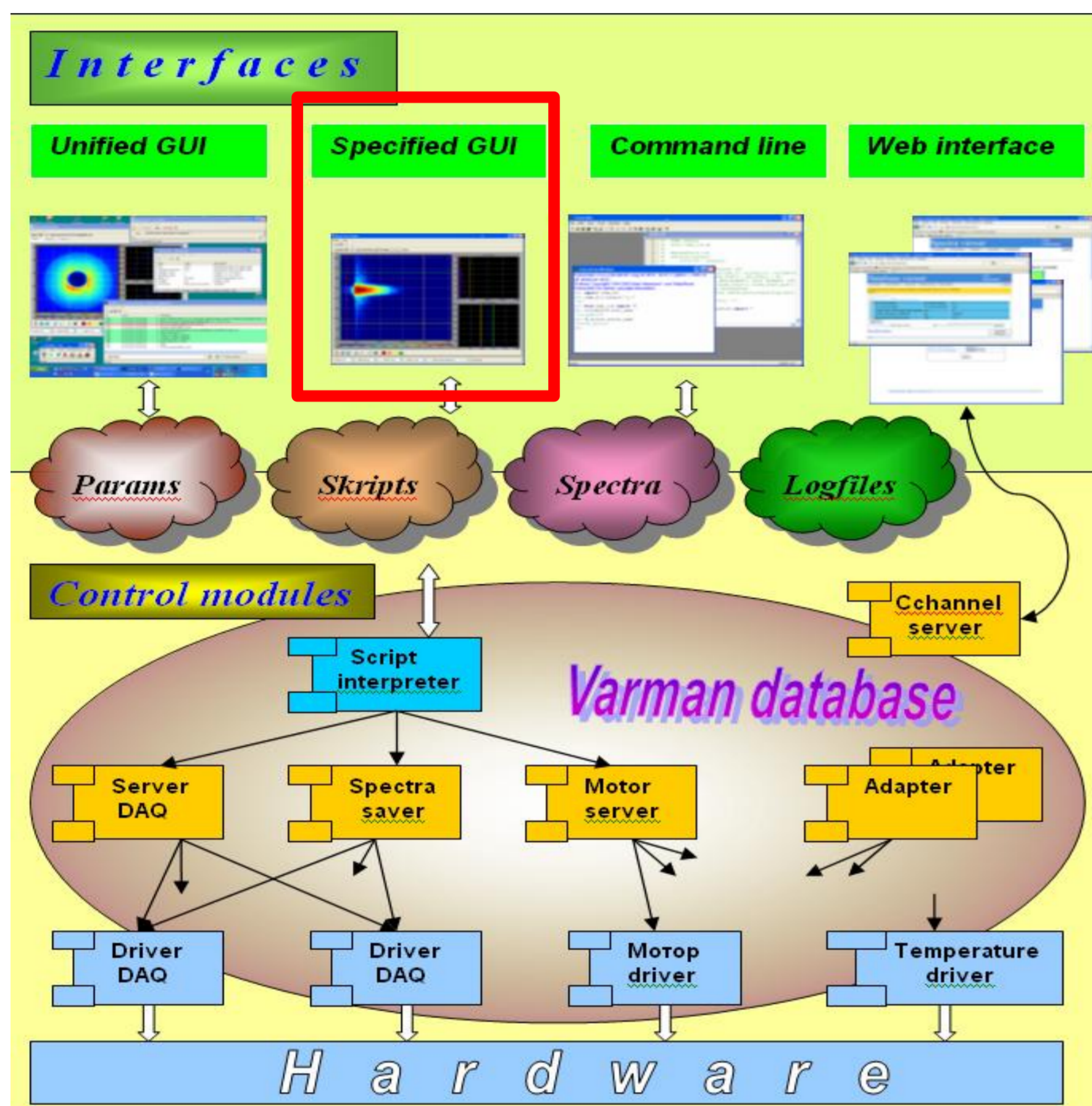


Figure 1. Sonix+ structure

## Past programs for adjusting spectrometers

Program<sup>2</sup> for the Epsilon spectrometer. The program:

- + structure was formulated well;
- was not completed.

Program<sup>3</sup> for the Remur spectrometer. The program:

- + has a friendly user interface;
- + has all necessary tools that are available;
- was developed under the Remur spectrometer;
- adaptation for other spectrometer is impossible.

## Development of new user programs

The suggested method lies in the following:

- presentation of each task in the form of a separate independent component;
- development of interface to the component for interaction with other components;
- combination of the components under the principle of «Mediator»<sup>4</sup> pattern in the form of large-scale components or programs.

## Use software tools

Main software tools:

- Python 2.6 – program language;
- PyQt – creation of the GUI;
- Matplotlib – Visualization;
- Numpy, Scipy – mathematical computations.

Additional software tools:

- *Eclipse* - integrated development environment;
- Utility *pyinstall* - combination of the Python files in the form of an exe-file.

## How do you create the components?

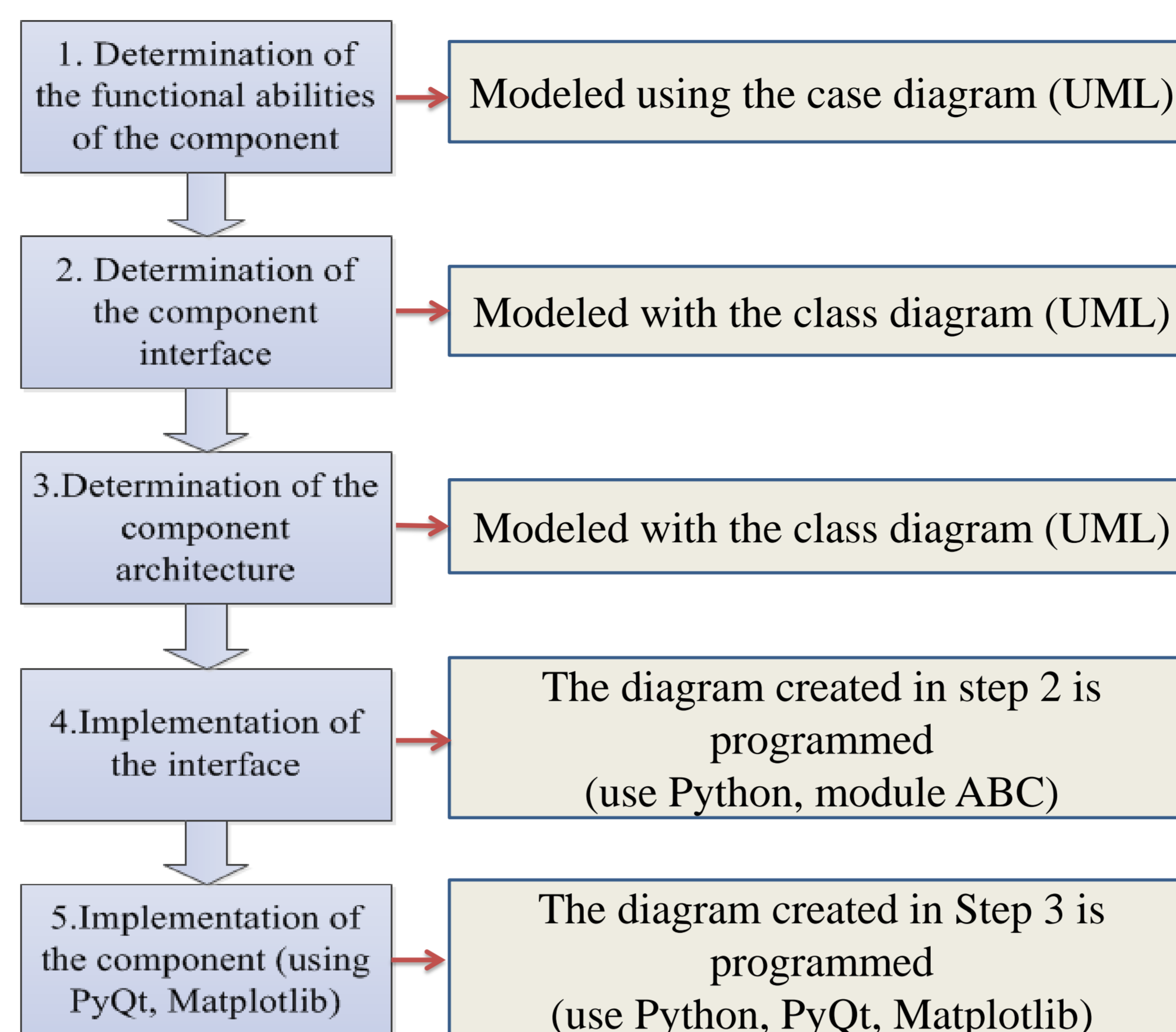


Figure 2. Algorithm for creation of components

## Created components

After analysis of the created adjustment programs<sup>2,3</sup> the main components necessary for the adjustment programs were differentiated. These are:

- “Measurement planning, performance and control” component– a list of measurements is created and performed;
- “Dependency visualization” component– represents the dependencies calculated on the basis of the spectra obtained;
- “Navigation” component – shows the created scans and the relevant spectra obtained in the course of measurement;
- Components for data visualization:
  - With point detectors;
  - With one-dimensional PSD;
  - With two-dimensional PSD;
- “Scanning and spectra” component – does not have a user interface. It is a database based on XML files;
- “Configuration” component – does not have a user interface. It obtains information from a XML file which contains the specification of the spectrometer. This file is created for each spectrometer

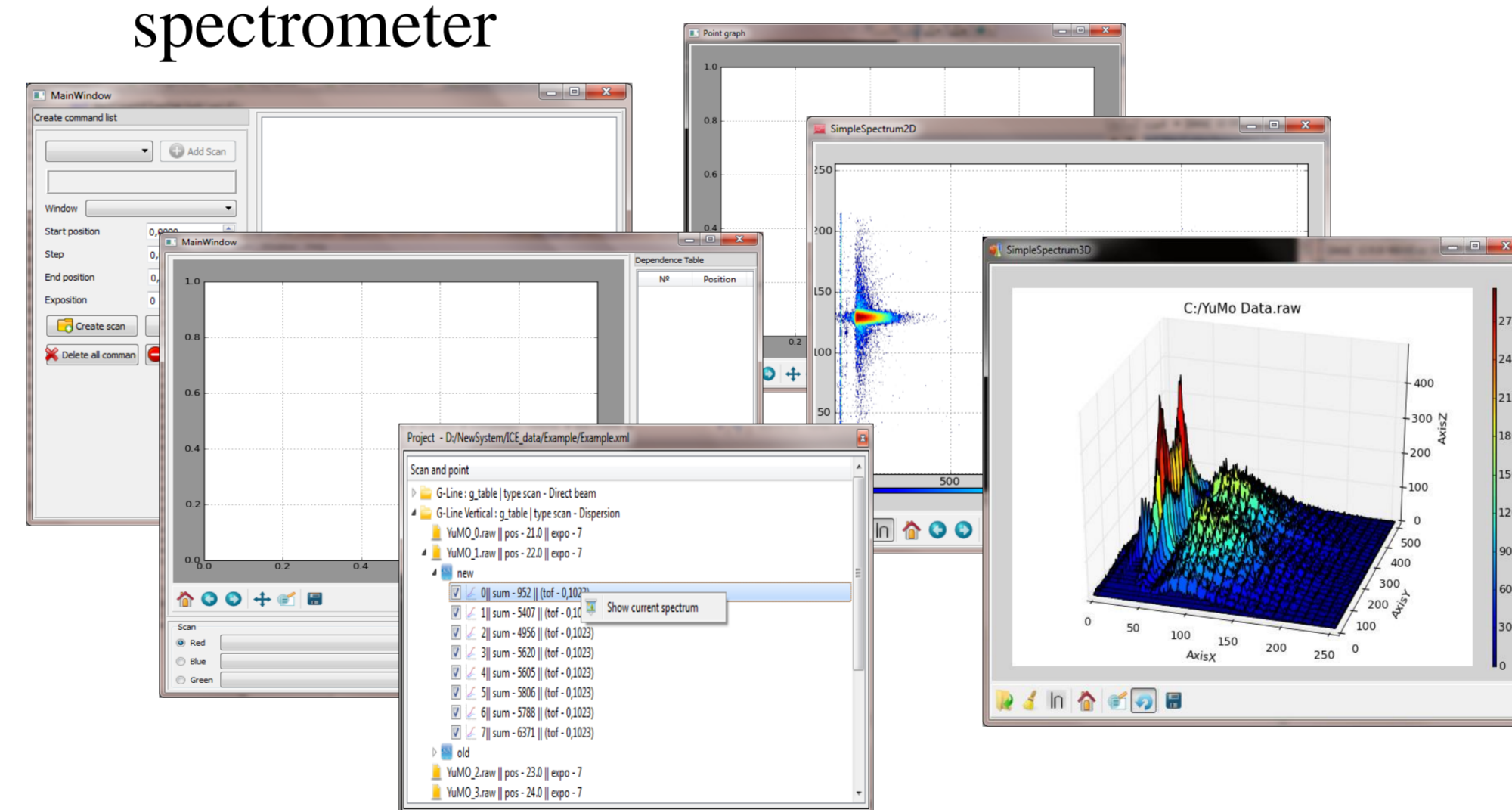


Figure 3. Ready components with user interface

## Developed architecture

Using the idea of the “Mediator” pattern and created components I developed architecture of a universal program for adjustment of the spectrometers «Remur», «YuMO» and «Reflex».

«Medator» component – accepts the inquiries between independent, not related objects and implements the set logic of interaction between them. All components don't know anything about each other. This creates weak connectivity of the system and enabling independent modification of the logics of interaction between them.

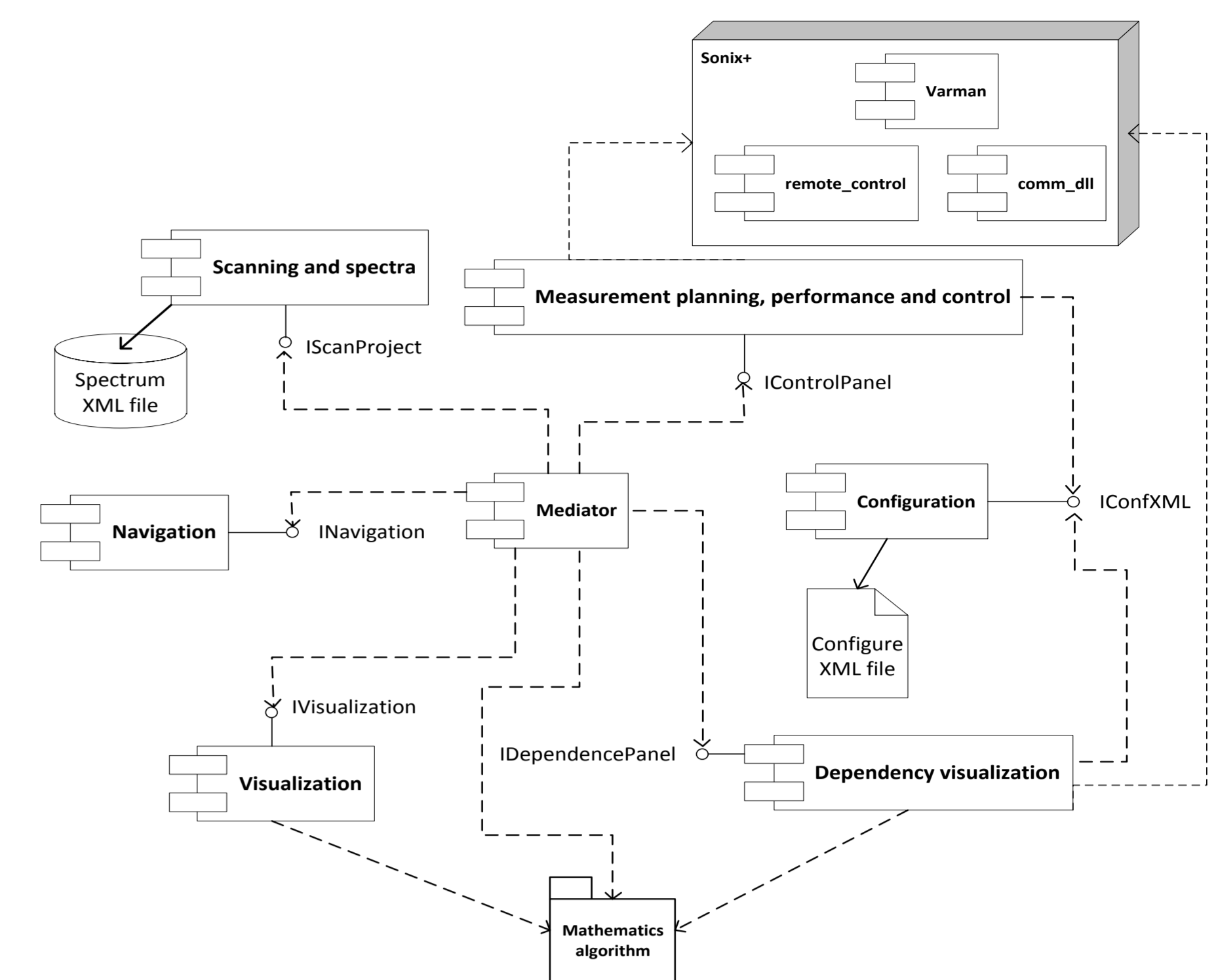


Figure 4. Architecture of the program for adjustment of the spectrometers «Remur», «YuMO» and «Reflex»

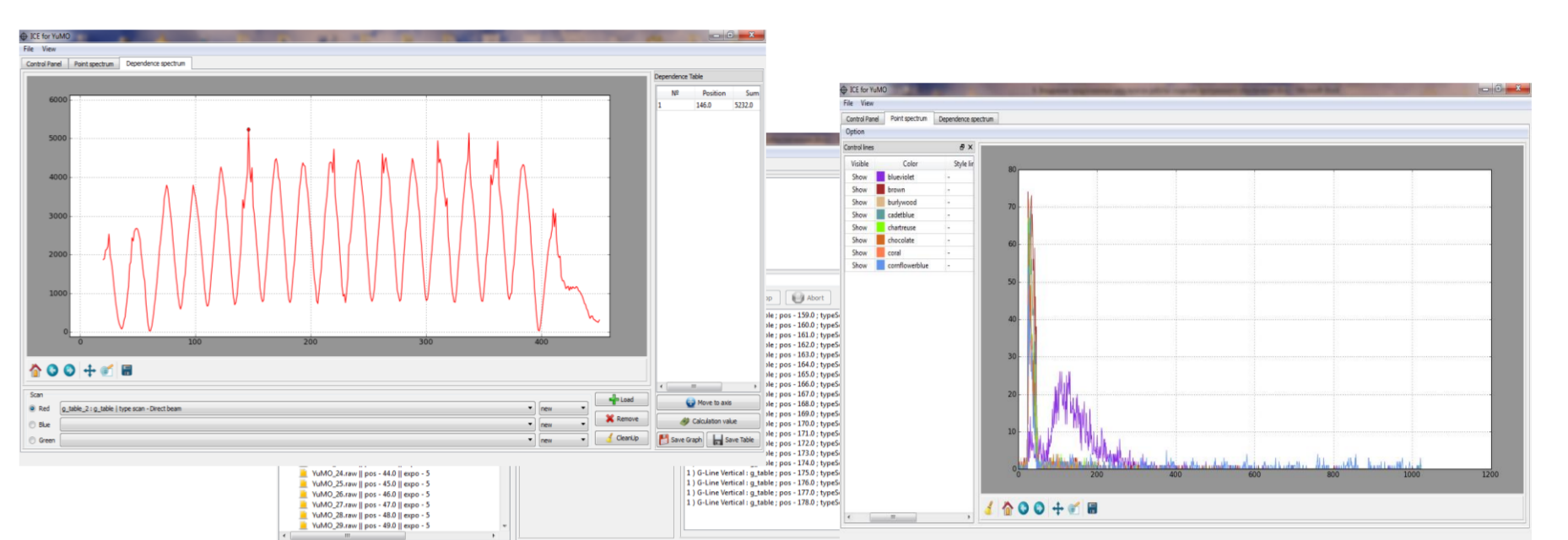


Figure 5. Example – program for adjustment for small-angle scattering YuMO

## Built program

Created components used for the development of a program for:

- adjustment of the spectrometers «Remur», «YuMO» and «Reflex»;
- visualization of the spectra with all spectrometers of the IBR-2 reactor;
- experiment control for the DN6 spectrometer;
- experiment control for the NPD<sup>5</sup> spectrometer established on the GEK-5 reactor, town of Obninsk, Russian Federation.

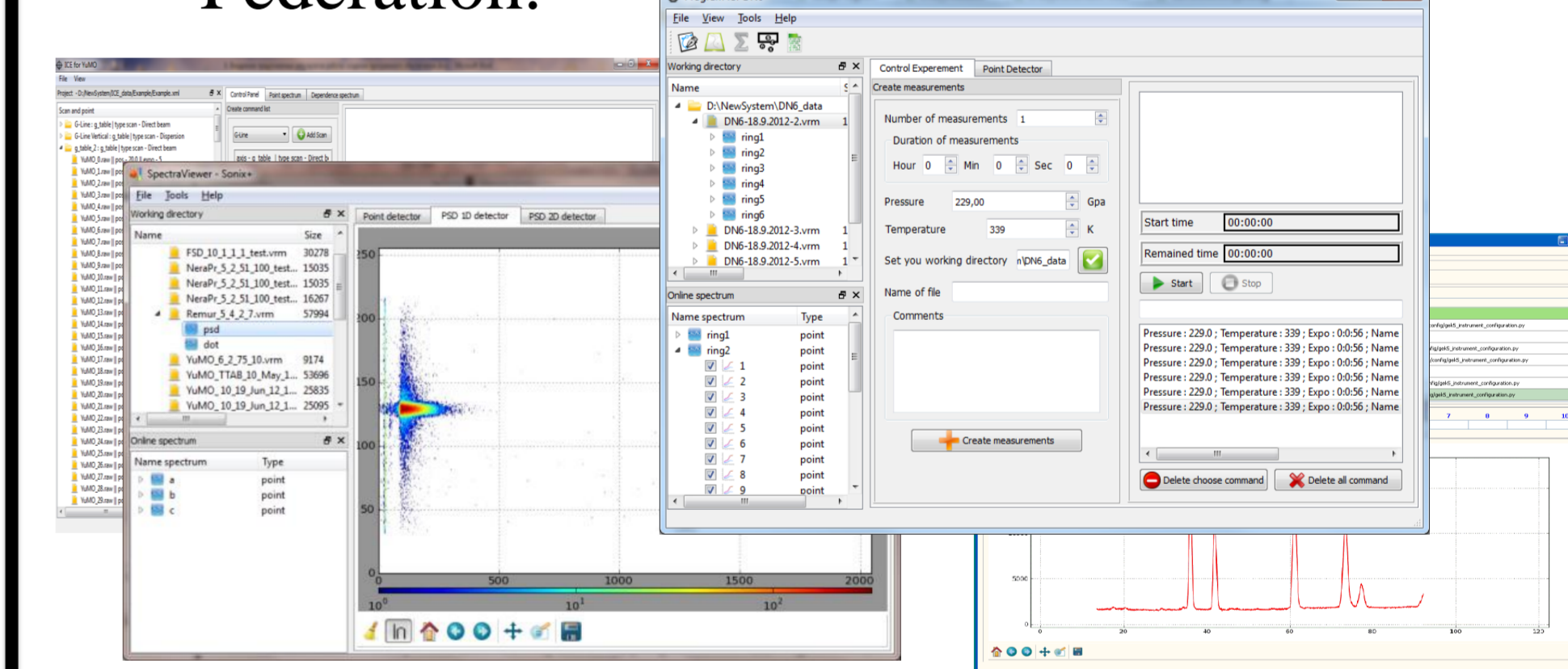


Figure 6. Build program

## Conclusions

- Shortening of the time for development of specialized programs;
- Fast creation of graphic visualization applications;
- Creation of new functionally independent components with GUI on the basis of «PyQt»;
- Simple combination of the created components into one program;
- Developing cross-platform programs.

## The future plans

- Create a library with components;
- Improving the method for combination of the components;
- Create development environment for combination of the components.

## References

1. Sonix+ - [www.sonix.jinr.ru](http://www.sonix.jinr.ru)
2. Astahova N.V., et al. Software complex for optimum adjustment of the diffractometer detector Epsilon: Preprint P13-2002-94. JINR, 2002. – p.11
3. Yudin V.E. Program for adjustment of the spectrometer REMUR in MS WINDOWS: Preprint P13-2003-12. JINR, 2003 – p. 10
4. Erich Gamma, et al. Patterns: Elements of Reusable Object-Oriented Software: Addison-Wesley 2001. – p.352.
5. Bogdzel A.A. et al. Experiment Automation System for a Neutron Powder Diffractometer: Preprint P13-2012-46. JINR, 2012. – p.11