DEVELOPMENT OF COMPONENT SYSTEM FOR NEUTRONS SPECTROMETRY AUTOMATION THROUGH THE USE OF NETWORK TECHNOLOGIES

Salamatin K.M.

International University "Dubna"
str. University 17, Dubna, Moscow reg., 141980 Russia

For experimental studies it is characteristic frequent changes in methods of research, which leads to the need to change the used hardware and software. From the literature it is known that the modification of experiment automation system (EAS) at best spent about 6 months of several programmers work. The reasons for such long periods of time: the complex nature of the work, the lack of an adequate strategy to code reuse and problems, arising at the attempt to transfer of functions of its modification and maintenance to experimenters.

The report is based on several works devoted to the technique to reduce the cost of development and modification of the EAS, increase system reliability and performance of experimental studies. The basic criteria of cost - time for development and modification of EAS. Code reuse is basic methodology, which is used to reduce labor costs in the development of complex systems. Therefore, in a task: to develop the architecture of the EAS, the method of its integration and algorithms providing reusability of developed code and transfer the function of modification to users.

The main reason for the loss of the ability to reuse – hard coupling of components and variability of the application logic. One can assume that the solution of these problems can be found by considering and adequate choice of advanced network technologies and taking into account of specificity of problem area.

Area of our interest is the automation of neutron spectroscopy at the IBR-2 reactor and accelerator IREN, so the arguments will be illustrated by examples from this field of spectroscopy. Let us consider the features of served area, which essentially affect on the solution of mentioned problems.

1. Analysis and classification of the functional components of the EAS for neutron spectrometry

EAS should solve the following tasks:

1) control of the environment of the sample surround;

2) registration and maintaining the flow of data from the detectors;

3) control of correctness of the experimental setup and the reliability of the recorded data;

4) presentation of data in a format suitable for processing.

The necessary step to ensure reusability - the separation of the EAS code to separate functionally complete modules, the purpose of which, for example, the following:

1. The user interface (multiple).
2. Data acquisition subsystem DAQ (input, conversion, and archiving).
3. Sample environment control.
4. Experiment control.
5. Monitoring the state of the object.
6. Fault processing.
7. Pre-processing of data.
8. Data visualization, etc.

Analysis of the functional components of the EAS [1] leads to the following conclusions:

• The number of software components of different functional types is small (tens).

• Requirements for data transfer rates are critical only for DAQ components, and this problem can be solved by the organization of common access to the memory.

• Any component in terms of computer science can act both as a client and as a server. However, in functional terms, these roles are not symmetrical, and we can identify the "control" and "executing" components. The role of control and executing components is predetermined by the logic of the experiment. The corresponding asymmetry is observed in the nature of messages that need to share components.

• It is possible to formulate two component groups of SAE:

a. basic components with deterministic nature of the interaction that implement the logic of the experiment; the nature of the interaction is one-to-one;
b. components (and some methods of basic components), implementing auxiliary logic - service functions, handling emergency situations, and others that do not affect (in normal conditions) performing of basic tasks; the nature of the interaction is one-to-many.

• Among the basic components there are only two types of clients, and as a result they require only a confirmation of the fact of execution for synchronization of operation. These are - user interface and experiment control program. In fact it is - the control software, and control - unidirectional. The implementation of the feedback refers to the auxiliary logic.

• Among the basic components there are only two types of services that perform the work defined by the experimental method. These are data acquisition subsystems (DAQ) and sample environment devices control components.

• There are three layers of software:

1. Control programs.
2. Programs which execute the actions required in the experimental procedure.
3. Programs of auxiliary logic.

• Most often the experimental techniques description and binding of components are done in many ways - as fragments of the system code, scripts, by the component technologies RMI, DCOM, CORBA, etc. These solutions lead to changes in programs or redundant code in components, slowing down execution, tight coupling, that impairs your ability to reuse the programs. No means found, what does not require to change component code or scripts or other part of SAE for describing method of experiment, that was not provided when EAS designed.

2. The choice of network technology for use in SAE

Methods for constructing of component applications most widely represented in networking. In order to reduce the amount of efforts, possible development errors and get the proven tools, performed an analysis of network technologies and a selection is made (with provision for specific problems of automation of experiments) of best options to use in the EAS [2]. The methods of RPC, technologies RMI, CORBA, DCOM, Ice, messages markup tools XML, JSON, SOAP, components search technology, architecture SOA were considered.

The above techniques have the following restrictions:

1. The classical RPC (client / server) is characterized with synchronous call, a way to interact one-to-one, and other properties that make complications in real-time systems.

2. RMI has a strict focus on Java.

3. DCOM may only be used in Windows.

CORBA more fully meet the requirements arising in the implementation of applications with often flexible logic. However, the major advantage of CORBA - a universal mechanism for dynamic binding - for EAS tasks excessively complex [3]:

- its implementation requires several hundreds of operators;
- require the inclusion of specific items in the environment to ensure the interaction of components;
- number of operations performed by the interaction, which is 40 times slower than the static binding;
- implements the scheme of interaction one-to-one.

Detailed gap analysis of CORBA can be found in [4]. A recent reincarnation of CORBA in Ice further complicates the dynamic binding scheme. In this paper we decided to perform development of dynamic components binding mechanism to take into account specificity of our tasks.

The optimal choice of the architecture is the use of SOA [5]. This architecture allows us to solve the problem of integrating application system with the ability to automatically change the composition of the components used in accordance with the evolution of the application logic and the change of the experimental procedure. However, SOA has not a mechanism for dynamic components binding.

Comparison of characteristics of JSON-RPC, XML-RPC and SOAP led to the decision to use JSON-RPC. The necessary support for dynamic search of components can be serviced by protocol SLP – the only received the status of the RFC.

Based on the analysis of the features of EAS and network technologies for component applications development concluded that there is need to solve two key tasks:

1. development of a universal mechanism for describing the experimental procedure that does not require code changes of ready components;

2. development of environment of components interaction, tailored to the specifics of the EAS and invariant to changes in the experimental procedure and auxiliary logic.

3. The proposed concept

Complex approach - consideration of the specificity of experimental studies, the use of modern technologies and the development of additional algorithms and methods - allows building a system, resting in division of programmers and users labor and providing possibility to modify system by user in accordance with current needed method of the experiment. Such concept includes the following provisions:

• The system is assembled from standard interacting functionally complete components providing a set of methods; components are represented in an executable format, they have an external interface – way to remote method call and response;

• code of the system components is invariant with respect to changes of experiment methods;

• experimental procedure is described with a unified interactive subsystem, invariant with respect to the experimental procedure and spectrometer design; the user interface of subsystem is based on the use of terms of the problem domain, not the operators of the programming language, or procedures names;

• integrating of the EAS according to the experiment method is carried out automatically;

• migration of components to other computers in the local network must not destroy the system or require any recompilation of the EPS components;

• execution sequence of functional components of managed types is controlled by the program, the algorithm of which is invariant with respect to the experimental procedure;

• experiment is performed in a fully automatic mode, with the ability to go to the dialog;

• the possibility of rejection of the use of software developers.

Development of interfaces, the control program, the management components of the sample environment does not contain ideological problems, a number of variants of these modules are tested in real experiments [6]. Consider the proposed solutions of two key tasks.

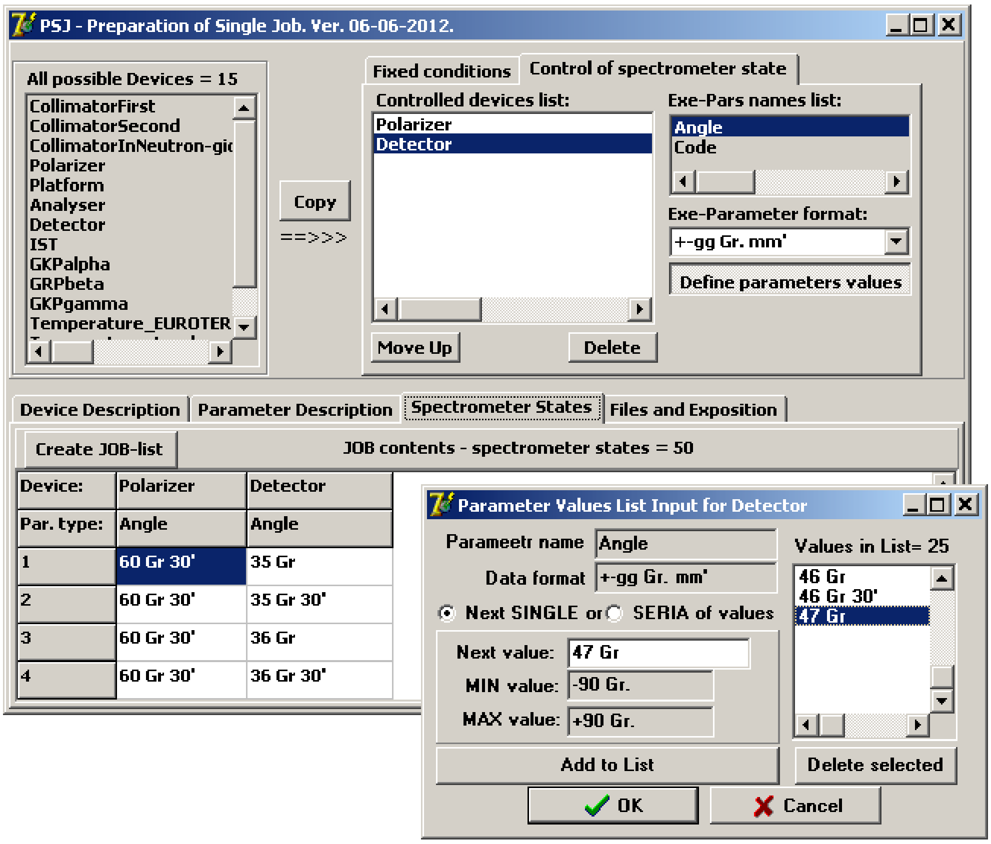

Fig.1. PSJ view window in edit mode.

4. Subsystem for descriptions of experimental techniques

Subsystem for description of experimental techniques include a database and two interactive programs: 1) program for making passports of components working with sample environment control devices which is used by programmers, and 2) job description preparation program PSJ (Preparation of Single Job), used by the experimenters.

4.1. Preparation of passports devices program (specialized text editor) creates a device description in JSON. The passport contains:

• the version of the passport;

• the name of the device;

• the type of component;

• a unique ID (GUID) to search component by the means of communication;

• a list of device parameters. Each parameter has a name, type, value, default value, and the range of acceptable values.

4.2. Job preparation program uses the list of available devices (and components) from the database. From this list by the dialog, the user selects components desired in this experiment. To these components during the experiment a several states will be given in accordance with the content of the job. Selecting one of the drives in this list, the user can specify a list of values for each control parameter. Based on these data PSJ creates a table, which presents an additional opportunity to change the composition and sequence of the components of the control equipment, and parameter values. PSJ view window in edit mode is shown in Fig. 1. The result of the PSJ is tabular description of a state machine that implements the necessary conditions in the experiment hardware system.

When describing the methodology the terminology of specific problem is used - the names of the spectrometer units, the angular positions, etc.

5. Distributed environment for serving components interaction

Components interaction environment is a key component of the EPS. Requirements for this component development are following:

1. asynchronous method invocation mechanism, because synchronous call slows down the speed of the application logic (and bandwidth), and complicates programming real-time systems, where the processes are carried out simultaneously, and events to be processed, there are asynchronous;
2. ability to convey information to several other processes;
3. one and the same environment interaction will handle all exchanges within the EPS, because homogeneous system is much easier to program and maintain;
4. transparency of interactions in a distributed system where any executable process must be able to communicate with any process in the system, using a single mechanism that is independent of the process placement;
5. a migration process from one machine to another, it will facilitate the elimination of emergency situations arising from failure of a computer;
6. interface should not depend on the boundaries of the computer - the code should be the same when referring to a process on the same machine or another, including a different type;
7. loss process, the computer or open the network connection should not lead to the destruction of the rest of the system;
8. automatic adjustment to your configuration.

In accordance with these requirements designed component DiCME (Distributed Components Messaging Environment).

Interaction environment is a link to the three layers of the programs;

1. Management programs.
2. The programs that do the work of the core logic.
3. Programs supporting logic.

One of the key characteristics of the interaction medium is a way of connecting components. For late (dynamic) binding information is provided by the program description of the experiment method. There is list of available devices (and components) from the database. User describes the methodology, using the name of controls surrounding the sample. Figure 2 shows the structure of the job file.

a) <GUID-1> <type> <method> <parameters list-1>
b) <condition-1> <condition-2> ... <condition-N>
c) <state-1> <state-2> ... <state-M>

Fig.2. The structure of the job file.

a) - the description of a condition
b) - the description of the state, in which data will be exposed
c) - the list of states - job file

Currently there are two types of devices - DAQ and CONDITION. Transmission method (SendMSG) is fixed, this method provides messaging executable components. List of parameters - part of the message, interpreted by code of component. The content of the list of parameters defined by application protocol and can be changed. For example, to control the position of the polarizer that list in JSON parameters format is as follows:

{device: "polarizer", parameter: "angle", value: "30grad"}

Specification line contains information about the service (GUID, type), enough to find it and link it always to one and the same component - control of the experiment component. To perform an action control program passes this line to service, using the medium of interaction. Figure 3 shows the algorithm of the control program and the way of implementing of basic logic SAE using DiCME. Experiment control program and DiCME are transparent for the parameter list.

We call this method of binding the " outer " as required for the binding and parameterization information is produced by means external in respect to the components to be bound. This method of binding does not limit the development of experimental techniques, represented as a finite state machine description (list of states SAE), and in all its changes do not affect the interaction environment.

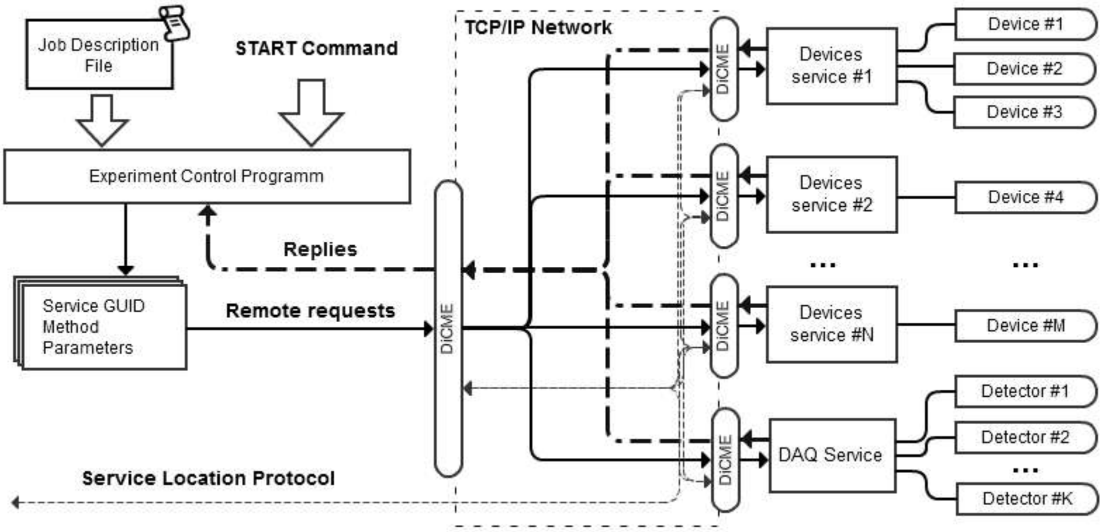

Fig.3. Algorithm of the control program and how to use DiCME for implementation of the basic logic of SAE.

The most significant difference auxiliary logic in respect to implementation of basic is the need to convey information to multiple processes, the list of which, generally speaking, to the source of information is unknown. To bind the auxiliary components we used method, in which the consumer once declare an interest in that type of information, then a dedicated server (Event Manager) serves all "signed" when a producer will publish this information. Fig. 4 shows the algorithm for maintenance auxiliary logic of SAE by DiCME.

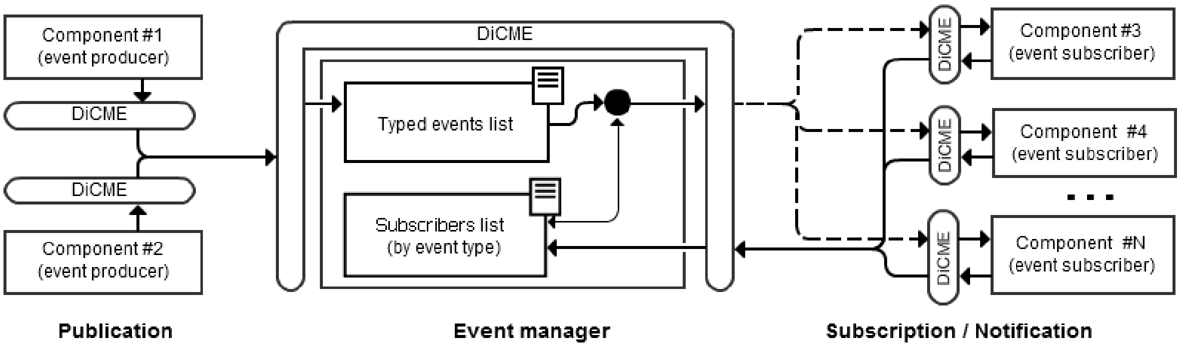

Fig.4. Algorithm DiCME when serving auxiliary logic SAE.

6. Conclusion

Finally, we present some of the distinctive features of this work:

• developed and implemented a new method of dynamic binding of components - an outer binding. Due to the outer binding it is not needed any alteration of the other SAE components when the base and auxiliary logic changed; the speed characteristics of the developed method do not yield possible with static binding;

• A new method and algorithms for experimental procedure describing are developed; universal subsystem for experiments techniques description implemented for spectrometry;

• classification of the components belonging to the base and auxiliary logic of the experiment introduced and used in the development of algorithms;

• used different mechanisms of a base and supporting logic realization;

• RPC mechanism is implemented through the message, followed by its interpretation in the body of the service component, thereby minimized components coupling and an asynchronous execution mechanism of action introduced;

• Search for network attached storage (NAS - Network Archive Service) is implemented using the same mechanisms as the search for other components;

• Developed DiCME component allows dynamically change the composition and addresses of basic and auxiliary components, which facilitates fault recovery in case of failure, improves the operation and management;

• Removed restrictions on the development of the logic of the experiment and the auxiliary logic without additional programming;

• SAE arrangement is done automatically; it allows modification of the EPS for new experiments by the user without the need for additional programming or compilation scripts;

• implemented the method of incremental EPS development in which the functionality of the EPS can be extended by adding new components without affecting other parts of the system;

• the communication layer developed is invariant under

- Changes in the basic and auxiliary application logic
- The emergence of a new type of message sources,
- The emergence of new event handlers.

Functions of communication messaging environment are written in general terms, not related to a specific area of application and can be used in various experiments and automation systems, automated control in other areas of concern.

A number of components tested in experiments on neutron scattering length of the noble gases [7], the series of experiments with polarized neutrons and targets at KOLHIDA instrument [8], experiments with a neutron source based on the meson factory at Red Pakhra [9], etc.

Work is done in accordance with the protocol on joint work LNP and Dubna University to develop programming techniques SAE.

Literature

1. V.N. Shvetsov, I.M. Salamatin, K.M. Salamatin. "The concept of building automation systems of neutron spectrometry of standard modules" // Preprint, to be printed, Dubna, 2013.
2. K.M Salamatin. "Selecting the technology of the component system for automation of experiments in the field of neutrons spectrometry" // Preprint, prepared for printing, Dubna, 2013.
3. http://kunegin.com/ref3/corba4/7.htm
4. Dubov. "On the road to SOA" // Chief Information Officer, № 8, 2005, p.12.
5. Michi Henning. "The Rise and Fall of CORBA" // ACM Queue, Volume 4, Number 5, June 2006; // http://citforum.ru/SE/middleware/corba_history.
6. V.N. Shvetsov, S.V. Alpatov, N.V. Astakhova, etc. "8-input system for TOF neutron-nuclear research-flight method" // PTE, 2012, № 5, p.54-61.
7. T.L. Enik, L.V. Mitsyna, V.G. Nikolenko et al. "Preparation for Testing Experiment on IREN Neutron Beam" // Proc. of Internat. Seminar "ISINN-17" (Dubna, May 27-29, 2009). Dubna: JINR, 2010. pp. 162-164.
8. J.G. Abov, V.P. Alfimenkov, L. Lason, etc. "Set 'KOLKHIDA' for experimental studies of interactions of polarized neutrons with polarized nuclei" // Preprint, P13-2008-69, Dubna, 2008.
9. V.L. Kuznetsov, E.V. Kuznetsova. "Study of the Effects of Parity Violation in Neutron Diffraction in Perfect Single Crystal" // Proc. of Internat. Seminar "ISINN-20" (Dubna, May 27-29, 2012). Dubna: JINR, 2013.