

P13-2013-92

К. М. Саламатин^{1,*}

**PSJ — УНИФИЦИРОВАННАЯ ПОДСИСТЕМА
ОПИСАНИЯ МЕТОДИКИ ЭКСПЕРИМЕНТА**

¹ Международный университет «Дубна», Дубна

* E-mail: del@tmpk.ru

Саламатин К. М.

P13-2013-92

PSJ — унифицированная подсистема описания методики эксперимента

Повторное использование кода — основная методология сокращения сроков разработки сложных систем. При попытке использовать компоненты системы автоматизации эксперимента (САЭ) в эксперименте с не предусмотренным заранее методом исследования или для другой экспериментальной установки мы сталкиваемся с необходимостью редактировать другие составные части САЭ. Объективных причин потери возможности повторного использования компонентов без изменений две: связанность компонентов и изменчивость методики эксперимента. В работе предложен вариант решения проблем, связанных с изменениями методики эксперимента, основанный на использовании выделенной подсистемы описания методики эксперимента и построения САЭ из типовых программных компонентов. Представлено обоснование решения, описание алгоритмов и реализации такой подсистемы.

Работа выполнена в соответствии с протоколом о совместных работах Лаборатории нейтронной физики им. И. М. Франка (ОИЯИ) и университета «Дубна» в области разработки методики программирования САЭ.

Сообщение Объединенного института ядерных исследований. Дубна, 2013

Salamatin K. M.

P13-2013-92

PSJ — A Unified Subsystem for Description of Experimental Technique

Code reuse is the main methodology that reduces the development time of complex systems. If you attempt to use the components of the experiment automation system (EAS) in an experiment with a research method which has not been provided, or with other experimental setup, we are faced with the need to edit other parts of the EAS. The objective reasons for the loss of the ability to reuse components are two. They are the hard coupling of components and variability of experimental techniques. In this paper we propose to solve problems related to changes in the experimental procedure, by the use of a dedicated subsystem for description of the experimental procedure and the construction of the EAS with standard software components. The reasons for this decision, a description of the algorithms, and implementation of this subsystem are presented.

The investigation has been performed in accordance with the protocol on the collaboration between the FLNP, JINR, and University «Dubna» on the development of the of EAS programming methods.

Communication of the Joint Institute for Nuclear Research. Dubna, 2013

Автоматизация работы экспериментальных установок существенна для результативного выполнения исследований. К настоящему времени созданы средства программирования и разработки систем автоматизации экспериментов (САЭ) различного масштаба — развитые системы программирования, средства быстрой разработки приложений RAD, системы типа SCADA, специальные средства построения систем управления экспериментальным и промышленным оборудованием [1]. Однако в ряде случаев сроки разработки САЭ оказываются не адекватными темпам и уровню развития аппаратной базы и средств программирования. Очевидный путь сокращения сроков — использование компонентных технологий построения САЭ с четким разделением кода системы по функциональному назначению. Существуют объективные причины, препятствующие повторному использованию компонентов, в числе которых связанность компонентов и изменчивость методики эксперимента. В [2] предложен вариант решения проблемы связанности компонентов. Изменчивость методики — неустраняемая особенность исследовательской работы. При возникновении необходимости выполнить эксперимент с использованием оборудования, не предусмотренного ранее в составе экспериментальной установки, приходится дополнять состав программного обеспечения, при этом включение нового компонента, как правило, приводит к необходимости внести изменения в другие составные части САЭ — программу или скрипты управления экспериментом, конфигурационные файлы и др.

Вариант САЭ [3], обеспечивающий решение проблемы изменчивости методики эксперимента, включает унифицированную подсистему описания методики эксперимента, типовые программные компоненты и среду их динамического связывания. В работе представлена реализация подсистемы описания методики и составления задания на эксперимент, а также модель унифицированной программы управления экспериментом, разработанные для экспериментов на реакторе ИБР-2 [4] и ускорителе ИРЕН [5] в области нейтронной ядерной физики и физики конденсированных сред.

1. УПРАВЛЕНИЕ ВЫПОЛНЕНИЕМ ЭКСПЕРИМЕНТА

Простейший способ управления работой САЭ — использование фиксированного интерфейса пользователя, специально разрабатываемого для конкретного спектрометра и позволяющего задать значения параметров исполняющей системе. В этом случае набор доступных пользователю методик опре-

деляется фиксированным кодом интерфейса, и выход за эти рамки требует изменения кода программ. Более гибким и часто используемым является вариант, при котором в управляющую программу вводится интерпретатор и нужная в эксперименте последовательность операций описывается программой на языке этого интерпретатора (скриптом). При этом система использует дополнительную информацию (адреса, константы и др.) из конфигурационных файлов. Практика показывает, что в 70 % случаев этот способ настройки содержит опечатки и может занимать до двух суток работы специалистов [6]. Помимо этого скрипты и конфигурационные файлы вносят связанность компонентов.

Рассмотрим *альтернативный вариант* программы управления основными задачами (базовой логикой) эксперимента. *Модель этой программы* проста, она инициирует выполнение двух абстрактных процессов: 1) формирование состояния установки (условий эксперимента); 2) регистрацию экспериментальных данных — и управляет их чередованием.

Для конкретизации каждого из указанных процессов программе управления нужны списки компонентов, которые выполняют формирование условий. Такая информация, дополненная значениями параметров, детализирующих условия эксперимента, является описанием методики эксперимента и может быть сформирована средствами, внешними по отношению к программе управления экспериментом.

Конкретную работу по формированию нужных условий эксперимента и регистрацию данных выполняют соответствующие компоненты. Для синхронизации последовательности работы процессов от каждого из них требуется только сигнал о завершении работы. Компоненты должны выполнять интерпретацию параметров, благодаря чему программа управления экспериментом (а также среда обслуживания взаимодействия компонентов) оказывается прозрачной для списка параметров, и это обеспечивает ей универсальность.

Предложенная модель программы управления экспериментом позволяет исключить вызов программы регистрации данных из описания задания на эксперимент и выполнять этот вызов по умолчанию. Этот вывод имеет существенное значение для разработки способа описания методики эксперимента.

Средства задания методики эксперимента необходимы (и присутствуют) в любой САЭ, однако реализация их в виде компонента ограничит область его применимости, а включение в виде функции в тело другого компонента внесет жесткую связанность. По этой причине в данной работе средства описания методики эксперимента выделены в специальную подсистему.

Подобный вариант с единым для различных САЭ интерфейсом пользователя и управляющей программой, настраивающейся на нужную методику эксперимента с помощью файла описания задания, использован нами в работах [3, 7].

2. АНАЛИЗ ОПЫТА ЭКСПЛУАТАЦИИ ПОДСИСТЕМЫ СОСТАВЛЕНИЯ ЗАДАНИЯ

В работе [7] подробно описана подсистема подготовки задания, в составе которой была база данных (БД) и программа подготовки задания PSJ (Preparation of Single Job). БД обеспечивала информационную поддержку для специалиста (экспериментатора), описывавшего конфигурацию драйверного слоя САЭ и составлявшего задание на эксперимент, и содержала:

- описания готовых компонентов драйверного слоя, доступных для использования;
- описание конфигурации (список компонентов) драйверного слоя разрабатываемой САЭ;
- таблицы параметров и другие данные.

Отметим следующие ограничения такого варианта, стимулировавшие выполнение разработки новой подсистемы.

- В БД заносилось явное описание конфигурации драйверного слоя конкретной САЭ. Такая фиксация вносит связанность, ограничивает гибкость САЭ и не является необходимой. Целесообразно выполнять динамическое подключение компонентов управления окружением образца к системе, используя соответствующие технологии и описание методики эксперимента.

- В состав описания управляемого заданием драйверного слоя (и в таблицу задания на эксперимент) включался компонент, выполнявший ввод и регистрацию экспериментальных данных (DAQ). Вызов этого компонента должен быть исключен из описания методики и выполняться по умолчанию по следующим причинам:

- 1) настройка такого компонента, способ использования, состав команд управления существенно иные, чем у компонентов управления условиями эксперимента. В связи с этим для работы с оборудованием ввода и регистрации данных должна формулироваться отдельная подсистема DAQ и для ее настройки использоваться специальная дисциплина;

- 2) состав DAQ фиксирован для спектрометра, во всех состояниях установки должен выполняться один и тот же процесс регистрации данных.

- БД была представлена бинарным файлом со структурой, определенной программистом. С целью обеспечения лучшей читаемости и масштабируемости следует использовать общепринятые форматы записи данных, например JSON (Java Script Object Notation), XML или БД (например, SQLite).

3. РАЗРАБОТКА ПОДСИСТЕМЫ ОПИСАНИЯ МЕТОДИКИ ЭКСПЕРИМЕНТА

Подсистема описания методики эксперимента включает БД и две диалоговые программы: 1) программу составления паспортов компонентов, рабо-

тающих с устройствами управления окружением образца, используемую программистами; 2) программу подготовки задания PSJ, предназначенную для экспериментов.

1. Программа составления паспортов компонентов создает описание (документацию) в формате JSON. Из ряда альтернатив (XML, JSON, SOAP и др.) выбран JSON по критериям простоты, достаточности, компактности и наличия средств поддержки в популярных языках программирования (C++, Delphi, C++ Builder и др.). Паспорт содержит следующие данные:

- 1) версию паспорта;
- 2) имя контроллера устройства;
- 3) уникальный ID (GUID), используемый для адресации компонента средствами коммуникаций;

```
{
  "version":1,
  "controller-id":"6F9619FF-8B86-D011-B42D-00CF4FC964FF",
  "controller-name":"Kolhida-Motors",
  "devices":[
    {
      "name":"polarizer",
      "parameters":[
        {
          "name":"angle",
          "type":"int",
          "range-from":0,
          "range-to":90,
          "default-value":45
        }
      ]
    },
    {
      "name":"analyzer",
      "parameters":[
        {
          "name":"angle",
          "type":"int",
          "range-from":0,
          "range-to":90,
          "default-value":0
        }
      ]
    }
  ]
}
```

Рис. 1. Пример паспорта устройства

4) перечень параметров устройства (не может быть пустым); возможно задание взаимоисключающих групп параметров. Каждый параметр имеет:

- имя (уникальное, неизменное);
- тип значения;
- значение по умолчанию (если задано, то используется в команде Init);
- диапазон допустимых значений.

Описание атрибутов параметра может быть опущено, однако не рекомендуется это делать, так как будет отключен контроль значений при вводе. На рис. 1 представлен пример паспорта устройства.

Для многокоординатных устройств (гониометры и др.) более удобно (и соответствует стилю задания на эксперимент) каждую ось представлять как отдельное устройство, даже в тех случаях, когда управление всеми осями выполняется через один контроллер.

Важные свойства данного подхода:

- разработка компонента, работающего с оборудованием, считается завершенной только после внесения документации в БД;
- реализация компонента возможна на любом языке для любой операционной системы;
- использование компонента возможно на любой ЭВМ локальной сети без фиксации этого адреса в конфигурационных файлах или коде программ;
- состав параметров (прикладной протокол работы с компонентом) свободен и определяется разработчиком компонента.

2. *Программа составления задания* использует список доступных устройств (и компонентов) из БД. Из этого списка пользователь в диалоге выбирает нужные в данном эксперименте компоненты и составляет два списка. Первый — список устройств, которым должны быть отправлены (однократно) команды инициации (и фиксации) состояния части осей многокоординатных устройств (например, гониометров), по которым в этом эксперименте не предполагается использовать сканирование, а также заказ на периодическую передачу информации о состоянии устройства и удержание значения заданного параметра. Второй список — список устройств, для которых в процессе эксперимента будут задаваться несколько состояний в соответствии с нужной методикой эксперимента. Последовательно выбирая названия устройств из этого списка, пользователь может задать для каждого список значений управляемых параметров. На основании этих данных PSJ создает таблицу, в которой пользователю предоставляется дополнительная возможность изменить состав и последовательность работы компонентов управления оборудованием, а также состав используемых значений параметров.

Вид окна программы PSJ в режиме редактирования показан на рис. 2.

Результат работы PSJ — табличное описание конечного автомата, реализующего нужные в эксперименте состояния аппаратной системы. В каждом состоянии выполняется экспозиция данных.

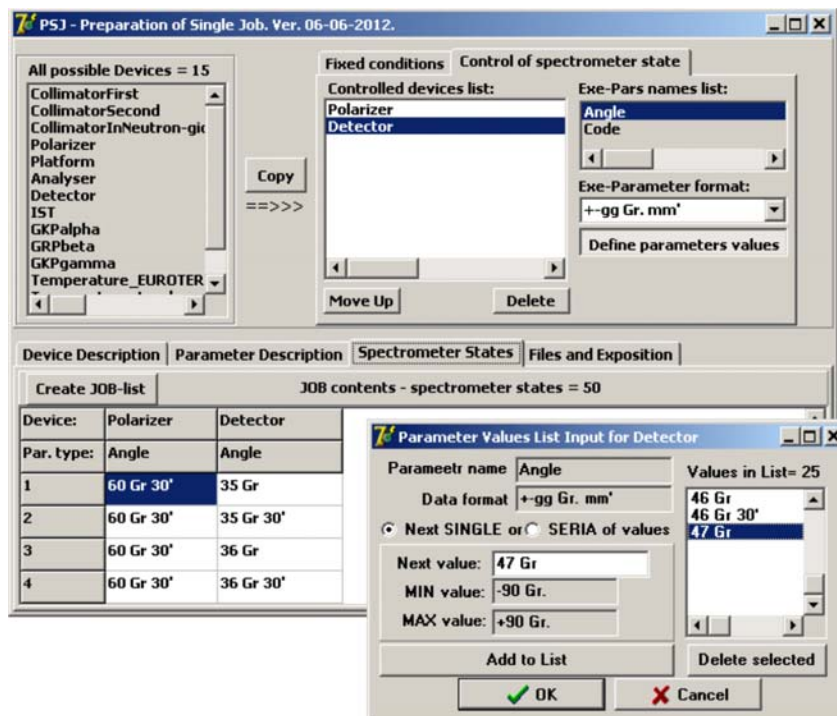


Рис. 2. Вид окна программы PSJ в режиме редактирования

На рис. 2 видно, что при составлении задания используется терминология конкретной области приложения — терминология экспериментатора: названия узлов его спектрометра, угловые положения и т. д.

ЗАКЛЮЧЕНИЕ

Разработаны алгоритмы, и реализована подсистема описания методики эксперимента, инвариантная относительно изменений методики исследования, конструкции экспериментальной установки и конфигурации САЭ. Сформулирована и использована модель унифицированной программы управления экспериментом, использующей файл задания на эксперимент. Свойства подсистемы заключаются в следующем.

1. Использование подсистемы существенно сокращает срок настройки САЭ для нового эксперимента.

2. Использование данной подсистемы и разработанной среды коммуникации DiCME [2] обеспечило возможность автоматически компоновать САЭ в соответствии с заказанной методикой эксперимента.

3. Подсистема и программа управления экспериментом пригодны для использования без изменений в различных САЭ.

4. Предложенный способ подготовки и исполнения задания сокращает объем необходимого программирования при разработке САЭ и одновременно увеличивает ее возможности. Без дополнительного программирования реализуется:

- возможность вмешательства в автоматическую работу САЭ с целью скорректировать содержание выполняемого задания;
- автоматическое определение точки входа для рестарта САЭ с известными и восстанавливаемыми потерями данных после сбоя питания, устраняемого отказа ЭВМ и др.

Эффективность предложения об использовании *подсистемы подготовки задания* подтверждается выполненными в последние годы подобными работами для систем на установках J-PARC/MLF [8] и ISIS [9]. Однако принципиальное отличие работ [8,9] в том, что в них не ставится задачи автоматической компоновки САЭ и динамического связывания компонентов: задание на эксперимент описывается скриптами и конфигурационными файлами, а последовательностью операций в эксперименте управляет интерпретатор. Недостатки такого подхода обсуждены выше.

Разработанная подсистема используется в реальных экспериментах на спектрометрах в Дубне [10,11]. Программная реализация подсистемы выполнена в общем виде и может использоваться при разработке САЭ для экспериментов в другой области приложения.

Работа выполнена в соответствии с протоколом о совместных работах Лаборатории нейтронной физики им. И. М. Франка ОИЯИ и университета «Дубна».

ЛИТЕРАТУРА

1. Kraimer M. et al. EPICS Input/Output Controller Application Developer's Guide. Release 3.14.8, 2005. // Argonne National Laboratory; <http://www.aps.anl.gov/epics>.
2. Саламатин К. М. DiCME — распределенная среда взаимодействия компонентов системы автоматизации экспериментов для физики низких энергий. Препринт ОИЯИ P13-2013-82. Дубна, 2013.
3. Саламатин И. М., Саламатин К. М. Разработка компонентной САЭ для физики низких энергий на основе использования сетевых технологий. Препринт ОИЯИ P13-2013-72. Дубна, 2013.
4. Драгунов Ю. Г. и др. Модернизация импульсного исследовательского реактора ИБР-2 // АЭ. 2012. Т. 113, вып. 1. С. 29–34.
5. Belikov O. V. et al. Physical Startup of the First Stage of IREN Facility // Proc. of Intern. Seminar «ISINN-17» (Dubna, May 27–29, 2009). Dubna: JINR, 2010. P. 10–16.

6. *Konnecke M. et al.* Treepath Based Instrument Control // NOBUGS2008, No. 132.
7. *Мазный Н. Г., Саламатин И. М., Саламатин К. М.* Генерация программ автоматизации экспериментов из модулей в формате загрузки. Препринт ОИЯИ Р13-2007-93. Дубна, 2007.
8. *Nakatani T. et al.* Present Status of the Computing Environment for the Experimental Instruments in J-PARC/MLF // NOBUGS2010, Preliminary Agendas NOBUGS40, 2010.
9. *Matt Clarke et al.* Making Life Easier for Scientists: Developing an Experiment Scheduler for the MLF // NOBUGS2010, Posters NOBUGS20, 2010.
10. *Enik T. L. et al.* Preparation for Testing Experiment on IREN Neutron Beam // Proc. of Intern. Seminar «ISINN-17» (Dubna, May 27–29, 2009). Dubna: JINR, 2010. P. 162–164.
11. *Абов Ю. Г. и др.* Установка КОЛХИДА для экспериментальных исследований взаимодействий поляризованных нейтронов с поляризованными ядрами. Препринт ОИЯИ Р13-2008-69. Дубна, 2008.

Получено 23 августа 2013 г.

Редактор *А. И. Петровская*

Подписано в печать 15.01.2013.

Формат 60 × 90/16. Бумага офсетная. Печать офсетная.

Усл. печ. л. 0,69. Уч.-изд. л. 0,83. Тираж 245 экз. Заказ № 58158.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: publish@jinr.ru

www.jinr.ru/publish/