

P13-2013-74

И. М. Саламатин, К. М. Саламатин^{1,*}

**РАЗРАБОТКА КОМПОНЕНТНОЙ САЭ
ДЛЯ ФИЗИКИ НИЗКИХ ЭНЕРГИЙ
НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ
СЕТЕВЫХ ТЕХНОЛОГИЙ**

Направлено в журнал «Прикладная информатика»

¹Международный университет «Дубна»

*E-mail: del@tmpk.ru

Саламатин И. М., Саламатин К. М.

P13-2013-74

Разработка компонентной САЭ для физики низких энергий
на основе использования сетевых технологий

Выполнен анализ специфики систем автоматизации экспериментов (САЭ). Показаны причины потери возможности повторного использования компонентов. Сформулирована концепция автоматически компонуемой САЭ, обеспечивающей использование компонентов в разных экспериментах без изменения других частей системы и модификацию САЭ пользователями. Определены ключевые задачи, решение которых необходимо для построения такой системы, и описано их решение с использованием современных сетевых технологий. Ряд разработанных типовых компонентов проверен в реальных экспериментах на реакторе ИБР-2.

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

Препринт Объединенного института ядерных исследований. Дубна, 2013

Salamatin I. M., Salamatin K. M.

P13-2013-74

Development of Component EAS for Low Energy Physics
on the Base of the Network Technology Usage

The analysis of specifics of the experiment automations systems (EAS) is performed. Reasons of the components reusability loss are shown. The conception of automatically integrated EAS, providing components usage in different experiments without change of the other parts of the system and modification EAS by the user is formulated. Key tasks, solution of which is required for building of such system are defined and their solution with use of modern network technologies is described. The group of designed standard components are checked in real experiments at the reactor IBR-2.

The investigation has been performed in accordance with the protocol about collaboration of the I. M. Frank Laboratory of Neutron Physics, JINR, and the university «Dubna».

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

Спектроскопия нейтронов является интенсивно развивающейся областью физики. Нейтронные исследовательские центры (сегодня их насчитывается несколько десятков) работают практически во всех развитых странах мира. Размеры этих центров различаются, количество нейтронных каналов варьируется от ~ 10 до нескольких десятков, примерно столько же используется различных спектрометров [1, 2]. Исследования ведутся в области ядерной физики, физики конденсированных сред, кристаллографии, медицины и др. Применяются все более совершенные системы и методы регистрации спектра излучения, изобретаются новые приборы и устройства, позволяющие с высокой точностью проводить измерения. 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 [3].

Однако в ряде случаев сроки разработки САЭ оказываются не адекватными темпам и уровню развития аппаратной базы и средств программирования. Объективная характеристика состояния дел с автоматизацией в СССР к 80-м гг. была дана в работе [4]. В этой работе выполнен анализ 200 САЭ по результатам опроса 40 различных организаций АН СССР и союзных республик, представлявших практически все направления, нуждавшиеся в автоматизации научных исследований. «На разработку и доведение каждой из 200 созданных систем автоматизации при наличии базовых средств затрачен в лучшем случае год, а в большинстве случаев до двух-трех лет. Для каждой системы программное обеспечение создается практически независимо» [4]. В работе [2] указано, что для уже работающих спектрометров при переходе к использованию ОС LINUX на разработку программного обеспечения 7 систем с использованием ранее освоенной разработчиками системы EPICS [3] израсходовано ~ 6 месяцев на систему. В связи с этим развитие методов, направленных на обеспечение преемственности разработанного кода и сокращение сроков разработки и модификации САЭ, остается актуальным.

Область интересов данной работы — развитие методики, сокращающей сроки разработки и модификации САЭ для экспериментов на реакторе ИБР-2 [5] и ускорителе ИРЕН [6] в области нейтронной ядерной физики и физики конденсированных сред. В работе сформулирована концепция САЭ,

автоматически компоуемой из готовых типовых модулей в исполняемом формате, разработана ее структура, необходимые алгоритмы и обоснована возможность реализации.

1. ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ ПРОГРАММ

1.1. Повторное использование — основная методология, которая применяется для сокращения трудозатрат при разработке сложных систем. Ее выгоды очевидны: уменьшается время разработки, возникает общность работы и в результате повышается качество продукта. Главный идеолог CodeGear (фирма Borland Software) Дэвид Интерсаймон в число существующих сегодня проблем включил разрозненность систем и команд разработчиков, отсутствие интеграции между ними, а также *острую нехватку стратегии повторного использования программного обеспечения (ПО)* [7].

1.2. Связанность компонентов системы — существенная причина, которая может ограничить возможность их повторного использования. Связанность может быть слабая и жесткая. Жесткая связанность возникает, если для конфигурирования системы и управления ее состоянием используется:

- объединение модулей посредством ссылок в коде;
- интерпретируемый язык (скрипты) с доступом к компонентам;
- технология интеграции, требующая описания и кодирования элементов интерфейса для доступа к новым компонентам и др.

Преднамеренное включение жесткой связи используется с целью оптимизации общей структуры приложения, сведения к минимуму различного рода избыточности и неэффективности компонентов. В результате связь между компонентами системы становится более тесной, количество критических взаимозависимостей растет. В этом случае сбой одного из компонентов с большей вероятностью приводит к потере работоспособности всей системы, и даже небольшие изменения могут привести к необходимости привлечения сторонних специалистов (программистов) для выполнения дополнительного кодирования.

Слабая связь компонентов-полуфабрикатов является краеугольным камнем архитектуры составных приложений. Связанность слабая, если компоненты связаны между собой лишь в степени, необходимой для обеспечения межкомпонентного взаимодействия. Компоненты взаимодействуют друг с другом, используя свойства и действия. Для удаленного выполнения операций можно использовать компоненты несколькими различными способами. Применение обмена сообщениями вместо вызовов методов дает преимущество в виде относительной независимости запрашивающего и отвечающего компонентов. Компоненты, работа которых основана на обмене сообщени-

ями, слабо связаны между собой и могут на основании содержания сообщений предпринимать то или иное действие.

1.3. Возможность тиражирования результатов этапов создания систем весьма сильно зависит от ряда факторов:

- совпадения направлений исследований и технической политики в области автоматизации в разных подразделениях организации;
- отсутствие стремления оставить все «как есть»;
- наличия финансовых ресурсов для достаточно быстрого внедрения нового решения в подразделениях организации, чтобы избежать морального устаревания новых технических решений.
- практической направленности и реальности — возможности реализации предлагаемых решений на уже имеющихся компьютерных и телекоммуникационных ресурсах.

Основным элементом тиражируемого решения может и должен стать комплекс средств программного обеспечения (желательно — и технического, как, например, в ЛИЯФ [8]), включая соответствующую документацию, методическое и технологическое обеспечение ввода новых систем и расширения (изменения) функциональных возможностей действующих.

2. КОНЦЕПЦИЯ САЭ, КОМПОНУЕМОЙ ИЗ ТИПОВЫХ КОМПОНЕНТОВ

Важная задача разработчиков — выбор наиболее подходящей архитектуры проекта, которая обеспечивала бы высокую надежность работы, приемлемую стоимость и сроки разработки, возможность повторного использования кода компонентов, легкость модификации проекта в будущем. Говоря о *модификации*, на сегодняшнем этапе следует требовать возможности выполнить ее *силами пользователя* (экспериментатора) при условии наличия готовых модулей. В данной работе предлагается концепция компонентной САЭ, включающая следующие положения:

- система собирается из взаимодействующих типовых *функционально законченных компонентов*, предоставляющих набор методов; компоненты представляются в исполняемом формате, для них *определен внешний интерфейс*;
- *код компонентов системы инвариантен* относительно изменений *методики исследования*; иначе говоря — компоненты могут использоваться без изменения в разных экспериментах и других системах;
- методика исследования и задание на эксперимент описываются с помощью *унифицированной диалоговой подсистемы*, инвариантной относительно методики и конструкции спектрометра; интерфейс пользователя подсистемы основывается на использовании *терминологии обслуживаемой проблемной области*, а не операторов языка программирования или названий программ;

- компоновка системы в соответствии с заданием на эксперимент выполняется автоматически;

- перемещение компонентов на другие ЭВМ в пределах локальной сети не должно разрушать систему или требовать перекомпиляции любых составляющих САЭ;

- последовательностью выполнения функциональных компонентов управляет типовая управляющая программа, алгоритм которой инвариантен относительно методики эксперимента;

- эксперимент выполняется в автоматическом режиме с возможностью перейти в диалоговый;

- возможно отторжение программного обеспечения от разработчиков и передача функций сопровождения системы и ее модификации пользователям.

Для определения задач, которые предстоит решить при разработке такой системы, рассмотрим особенности систем автоматизации экспериментов.

САЭ должна решать следующие задачи:

- 1) управления окружением образца в соответствии с методикой эксперимента;

- 2) регистрации, представления в формате, удобном для обработки, и сохранения потока данных от детекторов;

- 3) контроля корректности работы экспериментальной установки и достоверности регистрируемых данных.

Автоматизация этих задач существенна для результативного выполнения исследований. Для их решения могут быть использованы модули, функциональное назначение которых, например, следующее:

- интерфейс пользователя;

- управление выполнением эксперимента;

- подсистема регистрации данных DAQ (ввод в ЭВМ, преобразование, архивирование потока данных);

- настройка DAQ;

- управление условиями эксперимента — устройствами окружения образца;

- средства мониторинга состояния установки;

- средства передачи команд и сообщений;

- обработка отказов, визуализация, предварительная обработка данных.

Такой состав выбран на основании рассмотрения опыта других нейтронных центров [9–11], опыта разработок и эксплуатации САЭ в ЛНФ [12, 13] и пожеланий экспериментаторов.

Данный список не является полным, и не в каждой системе используются все перечисленные функции, но он соответствует в основном текущему состоянию представлений о САЭ в ряде организаций и позволит сделать выводы, необходимые для решения поставленной в работе задачи. Базовый состав

модулей включает интерфейс пользователя, программу управления экспериментом, подсистему регистрации DAQ, модули управления условиями эксперимента и средства передачи команд и сообщений. Подсистема регистрации и состав модулей управления устройствами окружения образца определяются конструкцией спектрометра, а в конкретном эксперименте — методикой эксперимента. Программы настройки подсистемы регистрации зависят от конкретной реализации контроллера ввода данных и технического задания.

Рассмотрим подробнее наиболее важные характеристики перечисленных модулей. Несколько замечаний относится ко всем модулям в этом списке:

- Любой компонент (и отдельные его методы) в плане информатики может выступать как в роли клиента, так и в роли сервера, выполняя функции управления, исполнения действий, публикации и обработки информации.

- Логикой эксперимента предопределены роли управляющих (интерфейсы пользователя и программа управления экспериментом) и исполняющих компонентов (компоненты управления окружением образца и подсистемы DAQ), схема их взаимодействия фиксирована. Для этих функциональных групп характерно требование гарантированной доставки команды и подтверждения завершения работы.

- В составе САЭ присутствуют другие функциональные пары — источники информации, вырабатываемой в процессе работы САЭ, и потребители этой информации. Источник может публиковать информацию различного типа (адреса экспериментальных данных, информацию о состоянии узлов установки, диагностические сообщения и др.). Потребителю требуется информация определенного типа, потребителей информации любого типа может быть несколько либо ни одного. Методика эксперимента не определяет схему взаимодействия источников и потребителей информации.

- Нет существенных оснований к тому, чтобы в тело кода клиента вводить информацию о конкретном связанном сервисе (например, адрес), и наоборот. Мы можем иметь дело со слабо связанными компонентами.

Разработка модуля и представление его в исполняемом формате для включения в САЭ не вызывает серьезных трудностей. Проблемы возникнут на пути интеграции модулей в компонентную систему.

Первая проблема связана с подсистемой регистрации данных DAQ. В эту подсистему входят модули ввода в ЭВМ, преобразования и архивирования данных. Для этих модулей (часто работающих на одной ЭВМ, но в разных адресных пространствах) требуется обеспечить предельную скорость обмена данными.

Вторая проблема заключается в том, что программа управления экспериментом, методика которого может быть изменена, не имеет информации о составе модулей управления параметрами окружения образца, реализуемых методах и значениях управляемых параметров. Схема работы программы управления экспериментом проста: 1) установить на образце необходимые

условия; 2) выполнить экспозицию данных; 3) преобразовать и архивировать данные; 4) все повторить с новыми условиями. Поскольку САЭ (в основном режиме) работает автоматически, должен быть введен специальный механизм, с помощью которого пользователем будет определен исполняемый метод и параметры для каждого используемого в эксперименте компонента управления окружением образца. В данной работе предлагается такую роль отвести подсистеме составления задания. Наличие такой подсистемы существенно упростит работу средств компоновки системы и связывания компонентов и в составе предлагаемой САЭ является обязательным, однако потребует разработки ряда дополнительных алгоритмов.

Рассмотрим подробнее влияние методики эксперимента на конфигурацию САЭ.

2.1. Влияние изменения методики эксперимента на САЭ. Диапазон возможных экспериментальных методик определяется:

- 1) используемой (фиксированной для спектрометра) детекторной системы совместно с остальными узлами спектрометра;
- 2) набором устройств окружения образца, определяющих условия регистрации данных.

Конкретная методика работы САЭ в каждом сеансе определяется в задании на эксперимент. Пользователем задаются продолжительность экспозиции и условия регистрации данных — названия устройств и значения управляемых параметров окружения образца.

На одном и том же спектрометре может быть использовано более одного детектора. Несмотря на многообразие детекторных систем, может быть введено конечное количество форматов регистрации сырых данных (например, регистрация событий и/или гистограмм, разреженных матриц [14]). Это может служить достаточной основой для построения унифицированного интерфейса к используемым в организации средствам регистрации данных (DAQ) и для разработки типовых программных модулей сжатия и архивирования данных.

Изменение методики эксперимента (задания) является основной причиной изменения используемой конфигурации оборудования спектрометра, которое управляет состоянием окружением образца (сменой мишеней, температуры, давления, коллиматоров и т.д.). Соответственно должен изменяться состав программных модулей, реализующих заданную методику. В нашем случае, благодаря наличию централизованных служб управления базовыми установками и их инфраструктурой [1], количество устройств управления окружением образца не превышает нескольких десятков [15, 16]. Для описания такого окружения образца может быть использована диалоговая программа.

2.2. Требования к скоростным характеристикам средств коммуникации между процессами. 1) *Передача данных.* Процессы взаимодействуют друг с другом путем передачи команд, сообщений и данных. Различные процессы

предъявляют разные *требования к скорости обмена данными*. Критическим в этом плане является взаимодействие процесса ввода экспериментальных данных с процессом архивирования. Максимально допустимый поток данных на входе системы (а в некоторых случаях и корректность результатов эксперимента) зависит от скорости выполнения этой пары процессов. В ряде случаев к этим процессам добавляется процесс сжатия данных (например, гистограммирование).

Таким образом, узким местом при разработке средств коммуникации между процессами является способ передачи данных от процесса ввода экспериментальных данных к процессам их сжатия и сохранения. Очевидно, для этих трех процессов желателен непосредственный доступ к данным. Для остальных процессов в большинстве случаев достаточна скорость, обеспечиваемая современными дисковыми накопителями и протоколом FTP.

2) *Передача сообщений*. Скорость передачи команд и сообщений зависит от используемого транспортного протокола и среды передачи данных и не является критическим параметром. Время передачи пакета 25 кбайт по протоколу UDP + время получения подтверждения составляет 400–700 мкс, что вполне достаточно для процессов, выполняемых в интересующих нас САЭ [17]. Использование протокола TCP/IP незначительно увеличивает эту оценку. Более важной является задача выбора механизма передачи, так как от этого зависит ряд свойств разрабатываемой системы. Рассмотрим этот вопрос более подробно.

2.3. Группы компонентов, реализующие базовую логику САЭ и вспомогательные операции. Содержание операций взаимодействия компонентов, входящих в состав САЭ, можно представить следующим списком:

- заказ на выполнение определенного метода (RUN);
- передача сигнала о выполнении метода или возникновении ошибки (DONE/ERROR);
- периодическая передача информации (NOTIFY);
- передача информации о событии, изменении состояния (NOTIFY);
- передача данных мониторинга состояния объекта (NOTIFY).

Приведенные в скобках названия операций являются условными и введены для ссылок по тексту. Операция вызова определенной функции (RUN) должна выполняться асинхронно. Периодическая передача информации используется, в частности, для подтверждения работоспособности компонента.

Приведенный состав операций взаимодействия компонентов может обслуживаться тремя типами сообщений, это:

- 1) заказ на выполнение определенного действия с указанием списка параметров (RUN);
- 2) ответ исполнителя с указанием результата (DONE/ERROR);
- 3) публикация информации (NOTIFY) с указанием ее характера.

Сгруппируем компоненты принадлежности к логике приложения. Будем считать базовым минимальный *набор компонентов, необходимый и достаточный для решения основных задач САЭ*. Можно сформулировать две группы компонентов САЭ:

1) базовые компоненты, решающие основную задачу эксперимента — получение экспериментальных данных; характер взаимодействия этих компонентов детерминирован и определяется базовой логикой САЭ.

2) компоненты (и, возможно, методы базовых компонентов), реализующие вспомогательную логику — сервисные функции, публикацию информации, обработку нештатных ситуаций и др.

Выполнение базовой логики САЭ не зависит (в штатных условиях) от состава и характера взаимодействия компонентов вспомогательной логики. Этот факт позволяет:

- в подсистеме описания методики эксперимента учитывать только базовые компоненты;
- использовать различные алгоритмы связывания компонентов базовой и вспомогательной логики.

В табл. 1 перечислены базовые компоненты САЭ и ассоциированные с ними способы взаимодействия. Помимо типов компонентов, указанных в табл. 1, в состав базовых входят средства обеспечения взаимодействия процессов. Взаимодействие этих компонентов может быть обеспечено одним универсальным интерфейсом для передачи текстовых *сообщений с указанием их типа*, и при изменении *количества* базовых компонентов изменения остальных программ, реализующих логику эксперимента, не потребуются.

Таблица 1. Базовые компоненты и способ их взаимодействия

Компоненты в роли клиентов	Способ взаимодействия	Компоненты в роли сервисов	Ответ сервиса
Интерфейсы пользователя	RUN	Программа управления экспериментом	DONE/ERROR
Программа управления экспериментом	RUN	Группа подсистем DAQ	DONE/ERROR
Программа управления экспериментом	RUN	Модули управления окружением образца	DONE/ERROR

В составе второй группы компонентов САЭ (табл. 2) компоненты-потребители информации одного и того же типа (например, визуализаторы) могут быть представлены в нескольких экземплярах (а также в нескольких вариантах), поэтому требуется передавать одинаковую информацию нескольким потребителям, количество которых может изменяться в процессе работы

спонтанно и независимо от базовой логики эксперимента. Взаимодействие всех компонентов этой группы также может быть обеспечено универсальным интерфейсом для передачи *текстовых сообщений с указанием их типа*. При изменении количества компонентов-источников информации, а также при появлении дополнительных экземпляров вспомогательных компонентов-сервисов (или компонентов нового функционального назначения) изменения программ, реализующих логику эксперимента, не потребуются.

Предлагаемое группирование компонентов, фиксация состава интерфейсов и способа их использования не ограничивают возможностей развития САЭ и являются основой для решения проблемы связанности, указанной в п. 1.2. Наличие в составе САЭ компонентов, реализующих вспомогательную логику, и их специфика должны найти отражение в структуре САЭ и алгоритмах среды коммуникации.

Таблица 2. Компоненты, информация и интерфейсы вспомогательной логики

Компоненты — источники информации	Характер информации	Периодичность	Функции компонентов — потребителей информации
Программа управления экспериментом	NOTIFY	По чтению строки задания	1) Интерфейс пользователя 2) протоколирование работы
Модуль архивирования данных	NOTIFY	По записи файла	1) Визуализация данных 2) предварительная обработка 3) интерфейс пользователя
Средства мониторинга объекта	STATE	Определяется в задании	1) Интерфейс пользователя 2) протоколирование работы ...
Источники периодической информации	TIMED	Фиксирована	1) Среда обслуживания взаимодействия компонентов 2) протоколирование работы
Средства протоколирования работы	STATE	По мере возникновения	Регистрация событий, обработка сбоев и отказов

Модуль задания методики эксперимента отсутствует в списках базовых и вспомогательных компонентов. 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.

3. ОСНОВНЫЕ АЛГОРИТМЫ РАЗРАБАТЫВАЕМОЙ САЭ

Разработка типовых модулей управления экспериментом, подсистемами регистрации, параметрами окружения образца и др. — необходимое, но не достаточное условие построения САЭ с типовыми компонентами. Для реализации такой САЭ требуется решить две ключевые задачи:

- разработать типовую подсистему описания методики эксперимента в виде задания на эксперимент, учитывающую возможность изменения состава используемых модулей управления окружением образца и не нуждающуюся в перетрансляции при изменении их состава;
- разработать механизмы обслуживания взаимодействия между процессами — обмен командами, сообщениями, данными.

3.1. Типовая подсистема описания методики эксперимента. Простейший способ управления работой САЭ — использование фиксированного интерфейса пользователя, специально разрабатываемого для конкретного спектрометра (например, [18]) и позволяющего задать значения параметров исполняющей системе. В этом случае набор доступных пользователю методик определяется фиксированным кодом интерфейса, и выход за эти рамки требует изменения кода программ. Более гибким является вариант представления программы управления экспериментом в виде скрипта, в котором кодируется нужная в эксперименте последовательность операций, однако практика показывает, что в 70 % случаев текст нового скрипта содержит опечатки. Помимо этого, скрипты вносят связанность компонентов.

Альтернативный вариант — это единый для различных САЭ интерфейс пользователя к подсистеме описания методики эксперимента и программа управления экспериментом, автоматически настраивающаяся на нужную методику.

Для описания методики и составления задания на эксперимент требуется информация об устройствах и работающих с ними компонентах. Рассмотрим два алгоритма получения этой информации:

- 1) хранить паспорта устройств в БД и выбирать из БД специальной диалоговой программой до запуска САЭ и начала эксперимента;
- 2) хранить паспорта устройств в обслуживающих их компонентах, динамически составлять список нужных активных компонентов (после запуска САЭ) путем поиска их по типу с помощью механизма обслуживания взаимодействия компонентов (описан ниже) и запрашивать эти данные у компонента специальной диалоговой программой до начала эксперимента.

Алгоритм составления списков активных компонентов по их типам присутствует в механизме обслуживания взаимодействия компонентов. Однако для реализации подсистемы описания методики эксперимента и формирования задания на эксперимент выбран первый вариант по следующим причинам:

- задание на эксперимент возможно подготовить заранее на любой ЭВМ без запуска САЭ;

- можно подготовить к выполнению пакет заданий;
- вариант лучше защищен от ошибок пользователя.

Разработанная подсистема описания методики эксперимента включает базу данных и две диалоговые программы: 1) программу составления паспортов доступных компонентов, работающих с устройствами управления окружением образца, используемую программистами; 2) программу подготовки задания PSJ (*Preparation of Single Job*), используемую экспериментаторами.

Программа составления паспортов устройств (специализированный текстовый редактор) создает описание устройства в формате JSON. Паспорт содержит следующие данные:

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

- 4) перечень управляемых устройств (не может быть пустым);

Для каждого устройства задаются:

- 4.1) имя (уникальное, неизменное);
- 4.2) список параметров (возможны взаимоисключающие списки);
- 4.3) значения по умолчанию (используются в команде Init);
- 4.4) диапазон допустимых значений и др.

Описание атрибутов параметра может быть опущено, однако не рекомендуется это делать, так как будет отключен контроль значений при вводе.

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

Программа составления задания PSJ использует список доступных устройств (и компонентов) из БД. Из этого списка пользователь в диалоге выбирает нужные в данном эксперименте компоненты. Последовательно выбирая названия устройств, пользователь может задать для каждого список значений управляемых параметров. Вид окна программы PSJ в режиме редактирования показан на рис. 1.

На рис. 1 видно, что при составлении задания используется терминология конкретной проблемной области — терминология физика: названия узлов его спектрометра, угловые положения и т. д.

Результат работы PSJ — файл с табличным описанием конечного автомата, реализующего нужные в эксперименте состояния аппаратной системы. В каждом состоянии выполняется экспозиция данных. Структура файла задания показана на рис. 2.

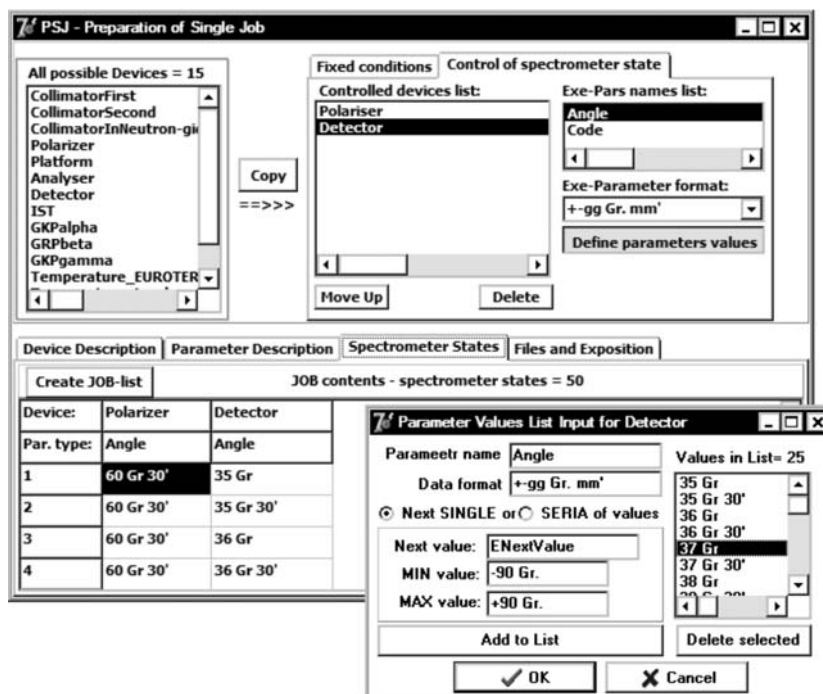


Рис. 1. Вид окна программы PSJ в режиме редактирования

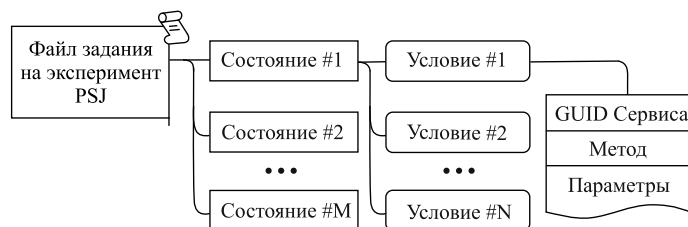


Рис. 2. Структура файла задания на эксперимент

3.2. Среда передачи команд и сообщений. Средства передачи команд и сообщений должны удовлетворять таким требованиям, как:

- соответствие международным стандартам;
- компактность, открытость, простота реализации, надежность;
- автоматический поиск и динамическое связывание компонентов;
- платформенная и языковая независимость;
- возможность выполнения удаленного вызова процедур в асинхронном режиме;

- возможность передачи информации одновременно нескольким потребителям;
- инвариантность относительно изменения логики приложения (базовой и вспомогательной);
- обеспечение безопасности.

На основании анализа, выполненного в работе [19], сделан вывод о целесообразности использования сервис-ориентированной архитектуры SOA, реализации взаимодействия на основе протокола JSON-RPC для передачи сообщений и SLP (*Service Location Protocol*) для поиска компонентов. В качестве транспортного протокола может быть использован HTTP (в конечном счете — TCP/IP). При использовании многозадачной операционной системы (Windows XP, Windows-7) механизм переключения задач может вносить во время передачи сообщения неопределенность, составляющую 20 мс. Это не повлияет на реализацию средств взаимодействия, так как процессы управления параметрами окружения образца, как правило, работают с электромеханическими устройствами со временем срабатывания (от $\sim 0,01$ с до нескольких минут), существенно превышающим время передачи команд и сообщений с использованием данного протокола [17].

Для осуществления взаимодействия разработана среда передачи сообщений DiCME (*Distributed Communication Management Environment*) в виде dll и функции API, включаемые линкером в компоненты для выполнения передачи команд и сообщений. На рис. 3 показана схема работы среды сообщений при обслуживании базовой логики.

Программа управления экспериментом (рис. 3) поочередно извлекает из файла задания описания нужных состояний экспериментальной установки, расчленяет их на описания условий, которые следует установить в окружении образца, и передает DiCME. Компонент DiCME по уникальному идентификатору (GUID) разыскивает исполняющий компонент и передает ему параметры, описывающие нужные действия. После установки всех нужных условий управляющая программа включает регистрацию данных. Программа управления экспериментом и DiCME прозрачны для списка параметров, это обеспечивает их универсальность и возможность использовать без изменения в различных САЭ.

Наиболее существенным отличием способа реализации вспомогательной логики от базовой является необходимость передать информацию несколь-

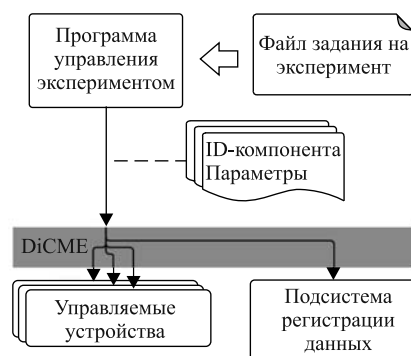


Рис. 3. Схема использования DiCME в САЭ

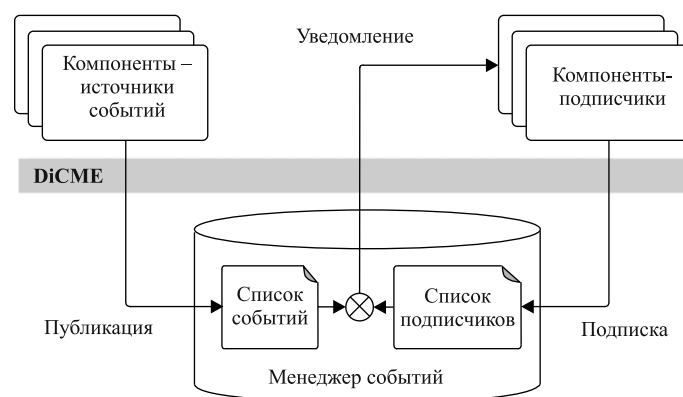


Рис. 4. Схема работы DiCME при обслуживании вспомогательной логики

ким процессам, состав которых, вообще говоря, источнику информации не известен. Для связывания вспомогательных компонентов выбран вариант алгоритма «подписки», при котором *потребитель* *однократно* *декларирует* *интерес к информации определенного типа*, после чего специальный сервер (Менеджер событий) обслуживает всех «подписавшихся» потребителей при появлении этой информации. На рис. 4 показана схема работы DiCME при обслуживании вспомогательной логики САЭ.

4. ДРУГИЕ КОМПОНЕНТЫ БАЗОВОЙ ЛОГИКИ

4.1. Интерфейсы пользователя. Интерфейс пользователя передает программе управления экспериментом команды оператора *START*, *PAUSE*, *CONTINUE*, *END*. Методы передачи этих команд относятся к базовой логике. В САЭ может присутствовать несколько интерфейсов пользователя одновременно, но активным может быть только один. Помимо операций базовой логики, в интерфейс могут быть включены функции вспомогательной логики (например, операции визуализации состояния системы, данных и др. — в случае специфического технического задания), однако универсальностью обладает вариант алгоритма, динамически составляющего список вспомогательных сервисов и предоставляющий возможность вызова любого из них.

4.2. Подсистема регистрации данных DAQ. Состав данной подсистемы DAQ следующий:

- диалоговая программа параметризации драйвера устройства ввода данных, программ преобразования информации и архивирования;
- группа компонентов ввода, преобразования и архивирования потока данных от детекторной системы.

4.2.1. *Обмен данными между компонентами DAQ и пропускная способность САЭ.* Выше было отмечено, что скорость обмена данными между компонентами DAQ является критическим параметром подсистемы. Передачу больших объемов данных между процессами в DAQ можно обеспечить путем предоставления им непосредственного доступа к одному и тому же участку оперативной памяти. Очевидно, это самый быстрый способ доступа. В ОС Windows такой общий ресурс реализуется в виде файла, отображенного в поименованный участок памяти. Работа с таким общим участком памяти организована двумя модулями:

- Модуль с функцией *GetSharedMem* (*SMemoryName:string*), которая возвращает адрес общего участка памяти *MemoryPTR*, зарегистрированного в ОС под именем *SMemoryName* (если он еще не существует, функция его создает). Данный модуль реализуется в виде dll.

- Модуль *SMemStruct*, описывающий структуру данных, адрес которой *MemoryPTR*.

Участок памяти, начинающийся с адреса *MemoryPTR* и имеющий структуру, описанную в модуле *SMemStruct*, мы будем называть ОПНД (Общая Память с Непосредственным Доступом). Для программ, которым требуется использовать общие данные, функция *GetSharedMem* и модуль *SMemStruct* включаются в тело программы линкером, и вызов *GetSharedMem* выполняется однократно при их инициализации. Всем программам, подключившимся к ОПНД, предоставляются одни и те же значения всех переменных, описанных в структуре данных. Любая программа может изменять значения этих переменных, используя семафор для устранения конфликтов при одновременном обращении к одной переменной.

ОПНД сохраняется до тех пор, пока к ней подключен хотя бы один пользователь (процесс). Благодаря этому во время работы САЭ различные программы могут загружаться и получать быстрый доступ к одному и тому же содержанию данных в ОПНД.

Функция *GetSharedMem* не зависит от конкретного варианта описания структуры данных (содержания *SMemStruct*) и может быть использована в различных проектах без изменения. Текущий образ ОПНД может сохраняться в файле и использоваться в процедурах быстрого автоматического восстановления работоспособности САЭ после сбоев и аварий.

Более сложная проблема возникает в случае, когда мощность (быстродействие + объем памяти) ЭВМ, на которой работает DAQ, недостаточна, чтобы в реальном времени выполнять последовательно ввод, преобразование и архивирование данных. В этом случае будет происходить потеря данных, и возникает необходимость решать проблему не только программными, но и аппаратными средствами — например, путем использования массива быстрых накопителей (RAID) и группы процессоров, параллельно обрабатывающих поток данных. Конкретный вариант решения этой про-

блемы зависит от многих факторов и в данной работе не может рассматриваться.

4.2.2. *Структура компонента ввода данных.* Примерная структура компонента ввода данных следующая:

- группа команд подключения к DiCME, которая обеспечивает взаимодействие САЭ с данным компонентом (6–8 операторов);
- пакет управляющих параметров;
- интерпретатор команд;
- реализация методов.

В отличие от других компонентов САЭ, компонент ввода данных после завершения экспозиции не шлет ответ вызывающей программе, а передает управление цепочке — преобразования и архивирования данных, и лишь после завершения архивирования вызывающей программе передается синхронизирующий сигнал разрешения продолжить работу.

Пакет управляющих параметров помещается в ОПНД, структура ОПНД должна включать описания:

- группы управляющих параметров для модуля ввода данных;
- таблиц, управляющих преобразованием (сортировкой, сжатием) данных;
- буферов спектров, заполняемых в реальном времени программами ввода, сортировки и др.;
- буферов для временного (до переноса в архив) хранения сегментов потока событий.

Стандартный набор выполняемых команд — *INIT*, *START*, *PAUSE*, *CONTINUE*, *END* — в комментариях не нуждается. Команда *INIT* выдается автоматически, остальные иницируются оператором САЭ, используя интерфейс пользователя.

В общем случае компонент ввода данных может работать со многими детекторами, при этом возможны варианты:

1. Один или группа детекторов с одинаковой структурой данных подключены через один интерфейс. Это самый простой случай, ниже (см. рис. 3) представлено окно программы параметризации такой подсистемы регистрации данных.

2. Детекторы с разной структурой данных подключены через один интерфейс. Очевидно, в этом случае просто увеличивается объем программы параметризации подсистемы регистрации.

3. Детекторные группы подключены через разные интерфейсы. Для каждого интерфейса, подключающего группу детекторов с иной структурой данных, потребуется дополнительный модуль ввода данных и соответствующие модули сжатия данных и параметризации интерфейса.

При работе с многодетекторными спектрометрами специальные меры для синхронизации времени старта детекторов обычно не нужны. Как правило,

начало и конец регистрации синхронизируется с фронтом синхроимпульсов от источника излучения. Помимо этого, задержка в программе времени передачи последовательных команд START используемым интерфейсам постоянна и не приводит к искажению данных. Если по условиям эксперимента такая синхронизация необходима (например, используется пространственно распределенная установка), то возможны два варианта:

- использование системы GPS при вводе данных от детекторов, не имеющих электрического соединения, дает возможность выполнить синхронизацию с погрешностью до 10 нс (метод подробно описан в работе [20]);
- если требуется синхронизация с погрешностью < 100 нс, то задача решается с помощью дополнительной электроники. Один из блоков (master) должен запрещать/разрешать прохождение синхроимпульсов от источника нейтронов к остальным блокам регистрации данных.

Продолжительность экспозиции обычно отсчитывается устройством регистрации данных.

Проиллюстрируем возможности конкретной подсистемы регистрации времяпролетных спектров [21] примером окна интерфейса пользователя к программе параметризации подсистемы. На рис. 5 представлено окно параметри-

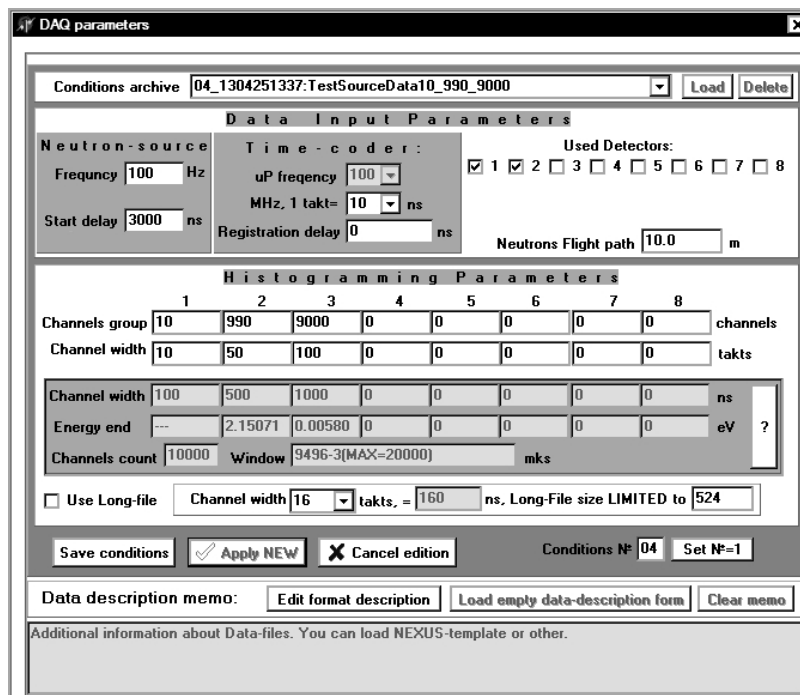


Рис. 5. Окно программы параметризации подсистемы регистрации данных

зации подсистемы, работающей с 8 детекторами, которые формируют данные с одинаковой структурой и подключены через один интерфейс. Пользователь получает возможность сообщить режимы работы источника нейтронов и временного кодировщика, подключить нужные детекторы (≤ 8), задать параметры гистограммирования (выделить несколько последовательных участков на оси номеров каналов, задать нужные ширины каналов) и др.

Значения параметров, заданных пользователем, помещаются в соответствующие поля структуры ОПНД, а также в архив.

На основании этих данных формируются также таблицы, управляющие алгоритмом гистограммирования. Использование ОПНД и разработанного быстрого алгоритма при сортировке данных в спектр с переменной шириной канала для окна в 20 мс при дискретности 10 нс позволяет за 20–30 нс выполнить обработку одного события для любого канала на ЭВМ с частотой процессора 2 ГГц.

Помимо задания значений параметров, управляющих процессом регистрации данных, окно программы предоставляет возможность ввести дополнительные данные, которые могут быть использованы для управления форматом представления данных и для математической обработки. После настройки подсистемы регистрации пользователю приходится изредка менять лишь формат гистограммирования.

4.3. Структура компонента управления окружением образца. Компонент управления устройством окружения образца содержит:

- группу команд подключения к DiCME;
- интерпретатор команд;
- процедуры, реализующие предусмотренные методы.

Протокол управления устройством (состав команд и параметров) определяется разработчиком компонента и документируется в паспорте устройства. Реализация обязательна только основного метода компонента, выполняющего перевод устройства в заказанное состояние. Состав остальных методов определяется конструкцией оборудования. После выполнения команды компонент шлет (асинхронно) вызывающей программе подтверждение ее выполнения или сообщение об ошибке.

В компоненты данного типа удобно поместить процедуры мониторинга состояния управляемого объекта и удержания заданного командой значения параметра.

Как указано выше, DiCME прозрачна для команд управления устройством, их интерпретация выполняется в компоненте.

4.4. Программа управления выполнением эксперимента. Основной алгоритм выполнения задания на эксперимент подробно описан в п. 3.2. Практика эксперимента требует некоторых добавлений:

- В данный алгоритм легко встраивается возможность пошагового выполнения работы, отключения регистрации или архивирования данных для

целей отладки и др. непринципиальные дополнительные операции, характерные для конкретной прикладной области, информация о которых в данной работе для краткости опущена.

- Обычной практикой прецизионных ядерно-физических экспериментов является расчленение многочасовой экспозиции в заданных условиях на несколько (обычно одинаковых) более коротких, результаты которых могут быть просуммированы. Для этого в задании указывается, сколько раз повторить все пункты задания (на рис. 6 — «цикл по проходам»). Благодаря этому, сравнивая данные в отдельных файлах, полученных при одинаковых условиях, можно обнаружить возможный дрейф фона, эффективности детекторов и др. эффекты и отфильтровать некорректные данные.

На рис. 6 представлена схема работы программы управления экспериментом.



Рис. 6. Схема работы программы управления экспериментом

Данная программа протоколирует работу: запоминает последнюю команду оператора, номер последнего завершенного состояния системы (выполненной строки задания) и некоторые др. данные и использует их для автоматического формирования названий файлов. Благодаря этому при сбоях возможно перезапустить систему — она продолжит работу, начиная с последнего незавершенного состояния.

5. СЕТЕВАЯ АРХИТЕКТУРА САЭ

На рис. 7 показана сетевая архитектура САЭ. Функциональные компоненты САЭ с целью обеспечения безопасности ограничены локальной сетью. Включение специальных средств обеспечения безопасности может ухудшить условия и скорость разработки САЭ, так как привлекаются третьи лица.

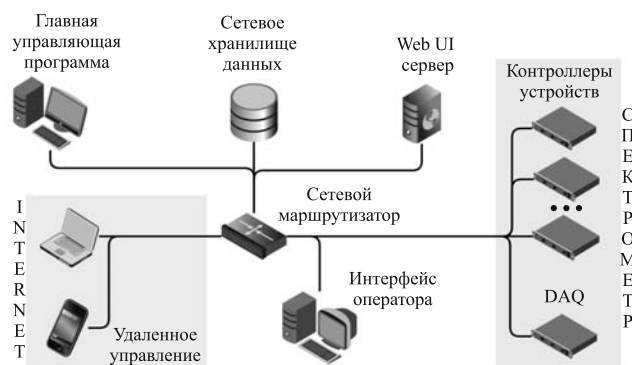


Рис. 7. Сетевая архитектура САЭ

В связи с этим было решено придерживаться идеологии доверенной сети. Все соединения и операции происходят без аутентификации (либо с паролями по умолчанию). Таким способом возможно избежать избыточной (не имеющей отношения к работе экспериментатора) параметризации системы — все компоненты находят друг друга сами, не задавая лишних вопросов. Тем не менее, при выборе протоколов и методов предусмотрена возможность будущего «упрочнения». Например, для транспорта запроса на выполнение метода вместо протокола HTTP может быть использован HTTPS, а авторизация с СУБД будет происходить по заранее установленной паре логин–пароль.

Во внешнюю сеть с соответствующими мерами фильтрации доступа вынесен WEB-интерфейс пользователя и дополнительные средства визуализации состояния САЭ и данных.

Для осуществления удаленного доступа через Интернет безопасность поднята на надлежащий уровень:

- интерфейс пользователя создан в виде WEB-фронтэнда к системе;
- специально настроенный фаервол дает доступ через Интернет только на соответствующие порты WEB-сервера;
- авторизация и работа возможна только по безопасному подключению (HTTPS).

ЗАКЛЮЧЕНИЕ

1. Предложенная концепция типовой САЭ, организация архитектуры и взаимодействия компонентов с использованием эффективных сетевых технологий и разработанные алгоритмы позволили:

- реализовать распределенную компонентную систему автоматизации экспериментов;

- динамически компоновать САЭ в соответствии с методикой, определенной заданием на эксперимент, силами пользователя без привлечения программистов;
- легко масштабировать САЭ (изменять количество ЭВМ, состав драйверного слоя и размещение компонентов);
- использовать компоненты, работающие под любой операционной системой, реализующей протокол ТСР/ІР;
- наращивать базовые и вспомогательные функции САЭ, не изменяя остальных программ;
- использовать компоненты без изменения в других САЭ.

Данная концепция направлена на сокращение сроков разработки и модификации САЭ, повышение надежности программ, в целом — на улучшение условий и эффективности экспериментальных исследований.

2. Предложена классификация компонентов, на ее основе разработан и впервые использован метод внешнего (по отношению к среде обеспечения взаимодействия) связывания компонентов. Этот метод позволил построить среду обслуживания взаимодействия, не зависящую от состава компонентов и не ограничивающую развитие логики САЭ. Исключена потребность в изменении готовых компонентов при изменении логики САЭ.

3. Предложенный способ обмена данными через ОПНД обеспечивает максимальную скорость. Подобная организация общей памяти системными средствами возможна и в других операционных системах [22], надежность и удобство работы с использованием ОПНД позволяет рекомендовать использование этого приема в других системах.

4. Впервые разработана подсистема описания методики эксперимента, инвариантная относительно изменений методики работы и конфигурации САЭ и пригодная для использования без изменений в различных САЭ.

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

Предложенный способ подготовки и исполнения задания сокращает объем необходимого программирования при одновременном увеличении возможностей. 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, устранимого отказа ЭВМ и др.

Данная подсистема принципиально отличается от популярного варианта управления процедурой исполнения эксперимента с помощью скриптов [23, 24]. В таких работах не ставится задача автоматической компоновки САЭ и динамического связывания компонентов, вместо этого информация об

адресах связываемых компонентов задается статически — в конфигурационных файлах и др. Такой способ настройки САЭ может занимать значительное время — до двух суток [25].

5. Разработанная программа управления экспериментом не зависит от методики эксперимента и конструкции спектрометра, реализует необходимые автоматический и пошаговый режимы работы, в том числе — характерные для прецизионной спектроскопии. Универсальность программы управления обеспечена разработанным методом внешнего связывания компонентов.

Универсальная подсистема составления задания, программа управления экспериментом, распределенная среда обеспечения взаимодействия компонентов САЭ и разработанные алгоритмы являются основными результатами работы. Разработанные компоненты и алгоритмы могут быть использованы другими организациями с иной проблемной ориентацией.

Ряд компонентов и подсистем [21], которые разработаны в соответствии с изложенными в данной работе принципами, в различных сочетаниях проверены опытной эксплуатацией в реальных экспериментах на спектрометрах в Дубне [6, 26, 27] и Красной Пахре [28].

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

ЛИТЕРАТУРА

1. *Ermiliv V. G., Ivanov V. V., Korolev V. S. et al.* The System for Diagnostics and Monitoring of the IBR-2 Reactor State. Data Acquisition, Accumulation and Storage of the Information // Proc. of the 2nd Intern. Workshop «DANEF-2000», Dubna, June 5–7, 2000. Dubna: JINR, 2001. P. 176–185.
2. *Levis P., Cooper G., Trouw F. et al.* Data Acquisition and Instrument Control at the Lujan Center: An Update. NOBUGS2010, Preliminary Agendas NOBUGS34, 2010.
3. *Kraimer M., Anderson J., Johnson A. et al.* EPICS Input/Output Controller Application Developer's Guide. Release 3.14.8, 2005. Argonne National Laboratory, <http://www.aps.anl.gov/epics>.
4. *Велихов Е. П., Выставкин А. Н.* Проблемы развития работ по автоматизации научных исследований // УСиМ. 1984. №4. С. 3–12.
5. *Драгунов Ю. Г., Третьяков И. Т., Лопаткин А. В. и др.* Модернизация импульсного исследовательского реактора ИБР-2 // АЭ. 2012. Т. 113, вып. 1. С. 29–34.
6. *Belikov O. V., Belozarov A. V., Becher Yu. 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. С. 10–16.
7. *Доррил Т.* Что нам готовит год 2027-й // PC Week/RE. No. 98 (614). 11–17 марта 2008 г.

8. *Соловей В. А., Савельева Т. В., Вихарев Л. Е., Колхидашвили М. А.* Базовые аппаратные средства для обеспечения интегральных и времяпролетных методов измерения в экспериментальных исследованиях конденсированного состояния сред с применением нейтронов. Препринт ЛИЯФ 2599. ЛИЯФ, 2005.
9. NOBUGS Conferences. <http://www.nobugsconference.org/>
10. International Symposium on Nuclear Electronics and Computing. <http://nec2011.jinr.ru/>
11. Сайт ПИЯФ. <http://www.pnpi.spb.ru/>
12. Proc. of the 1st Intern. Workshop «DANEF'97». Dubna: JINR, 1997.
13. Proc. of the 2nd Intern. Workshop «DANEF-2000». Dubna: JINR, 2001.
14. *Ji-Yong So.* Event Mode Data Reduction & Analysis Method for the Time-of-Flight Spectrometer // NOBUGS2010. No. 24.
15. *Куклин А. И., Сиротин А. П., Кирилов А. С. и др.* Автоматизация и окружение образца модернизированной установки ЮМО. Препринт ОИЯИ P13-2004-77. Дубна, 2004.
16. *Аксенов В. Л., Жерненко К. Н., Кожевников С. В. и др.* Спектрометр поляризованных нейтронов РЕМУР на импульсном реакторе ИБР-2. Сообщение ОИЯИ D13-2004-47. Дубна, 2004.
17. *Riedel R., Zolnierczuk P., Parizzi A., Sundaram M.* UDP, TCP, Circular Buffers, Multi-Threaded Programming and the Transmission of Event Data // NOBUGS2010. No. 47
18. *Astachova N. V., Kirilov A. S., Salamatin I. M.* Remote Control of the YUMO Spectrometer and User Interface // Proc. of the 2nd Intern. Workshop «DANEF-2000», Dubna, June 5–7, 2000. Dubna: JINR, 2001. P. 275–278.
19. *Саламатин К. М.* Выбор технологии построения компонентной системы для автоматизации экспериментов в области спектроскопии нейтронов. Препринт ОИЯИ P13-2013-72. Дубна, 2013.
20. *Астахова Н. В., Бордюгов Л. Г., Герасимов А. В. и др.* Распределенная беспроводная система регистрации с синхронизацией потоков данных. Препринт ОИЯИ P13-2006-41. Дубна, 2006.
21. *Швецов В. Н., Алпатов С. В., Астахова Н. В. и др.* 8-входовая система TOF для нейтронно-ядерных исследований по методу времени пролета // ПТЭ. 2012. № 5. С. 54–61.
22. http://habrahabr.ru/blogs/nix_coding/55716/
23. *Nakatani T., Inamura Y., Ito T. et al.* Present status of the computing environment for the experimental instruments in J-PARC/MLF // NOBUGS2010, Preliminary Agendas NOBUGS40. 2010.
24. *Matt Clarke, Takeshi Nakatani, Yasuhiro Inamura et al.* Making life easier for scientists: developing an Experiment Scheduler for the MLF // NOBUGS2010, Posters NOBUGS20. 2010.
25. *Konnecke M., Hauser N., Franceschini F. et al.* Treepath Based Instrument Control // NOBUGS2008. №132.

26. *Enik T. L., Mitsyna L. V., Nikolenko V. G. 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.
27. *Абов Ю. Г., Алфименков В. П., Ласонь Л. и др.* Установка КОЛХИДА для экспериментальных исследований взаимодействий поляризованных нейтронов с поляризованными ядрами. Препринт ОИЯИ Р13-2008-69. Дубна, 2008.
28. *Kuznetsov V. L., Kuznetsova E. V.* Study of the Effects of Parity Violation in Neutron Diffraction in Perfect Single Crystal // Proc. of Intern. Seminar «ISINN-20», Dubna, May 27–29, 2012. Dubna: JINR, 2013.

Получено 23 июля 2013 г.

Редактор *М. И. Зарубина*

Подписано в печать 03.10.2013.

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

Усл. печ. л. 1,69. Уч.-изд. л. 2,06. Тираж 245 экз. Заказ № 58075.

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

E-mail: publish@jinr.ru

www.jinr.ru/publish/