

УДК 004.415.2  
На правах рукописи

Мазный Никита Геннадиевич

Использование метода конечных автоматов для  
разработки программного обеспечения автоматизации  
экспериментов  
в области физики низких энергий

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата физико-математических наук

Дубна - 2009

Работа выполнена в НПЦ “АСПЕКТ”, г. Дубна.

Научный руководитель: кандидат физико-математических наук  
Саламатин Игорь Маркович, ОИЯИ, г. Дубна.

Официальные оппоненты: доктор физико-математических наук, профессор  
Никитин Владимир Алексеевич, ОИЯИ,  
г. Дубна,  
доктор технических наук, профессор  
Тюрин Владимир Федорович, ИТМ и ВТ  
им. С.А. Лебедева, г. Москва.

Ведущая организация: ГНЦ РФ Физико-Энергетический институт,  
г. Обнинск.

Защита диссертации состоится “ \_\_\_\_ ” \_\_\_\_\_ 2009г. в “ \_\_\_\_ ” часов  
на заседании диссертационного совета Д720.001.04 в Лаборатории  
информационных технологий Объединенного института ядерных исследований, г.  
Дубна Московской обл. 141980 Жолио-Кюри, 6.

С диссертацией можно ознакомиться в научной библиотеке Объединенного  
института ядерных исследований.

Автореферат разослан “ \_\_\_\_ ” \_\_\_\_\_ 2009 г.

Ученый секретарь диссертационного совета,  
кандидат физико-математических наук

З.М. Иванченко

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**АКТУАЛЬНОСТЬ работы.** Информационные и производственные технологии (информационно-телекоммуникационные системы, гибкие производственные системы, системы автоматизированного управления и др.) относятся к приоритетным направлениям науки и техники, и их развитие актуально не только в нашей стране. К настоящему времени созданы средства программирования и разработки систем автоматизированного управления (САУ) различного масштаба - развитые системы программирования, средства быстрой разработки приложений RAD, системы типа SCADA, специальные средства построения систем управления экспериментальным и промышленным оборудованием крупных объектов (например, EPICS, OPC). По мере развития средств разработки САУ увеличивается их сложность, возникает возможность автоматизировать управление более сложными объектами, улучшаются эксплуатационные характеристики разрабатываемых САУ. Одновременно растет число разработчиков, вовлекаемых в разработку близких по назначению САУ, разработка САУ приобретает комплексный характер и требуется привлечение специалистов разного профиля. В результате в ряде случаев сроки разработки САУ могут оказаться не адекватными темпам развития аппаратной базы и средств программирования.

Способ разработки САУ существенно зависит от их масштабов и области применения. Так, создание заводов, технологических линий по производству специализированной электроники, привлечение крупных коллективов программистов приемлемо для проектов национального масштаба, а также сложных или уникальных объектов автоматизации. Соответствующее программное обеспечение таких систем обычно довольно консервативно (срок жизни >10 лет) и для него решаются вопросы компетентного сопровождения. Сроки его разработки (>1..2 лет) адекватны срокам создания объектов и являются приемлемыми.

Данная работа посвящена вопросам разработки программного обеспечения (ПО) САУ меньшего масштаба, а именно: автоматизированных систем научных исследований (АСНИ) в области физики низких энергий, систем сбора данных для систем мониторинга, таможенного контроля, автоматизации технологических процессов в небольших научно-производственных центрах и т.п. Такие системы имеют свою специфику:

- автоматизируемые объекты следует квалифицировать как объекты небольшой сложности, с числом управляемых компонентов обычно в пределах нескольких сотен;
- системы подлежат тиражированию с небольшими изменениями, обусловленными требованиями заказчика или характеристиками автоматизируемого объекта;
- рассматриваемый класс САУ подвержен систематическому изменению методики исследования, состава контролируемых и управляемых параметров,

ввиду этого неоднократно изменяется состав используемого специального оборудования в течение срока жизни системы.

В этом случае роль сроков разработки ПО, тем более - адаптации программной системы к новым условиям использования, существенно возрастает. Системы автоматизации научных исследований, мониторинга, таможенного контроля и т.п. имеют между собой много общего, поэтому далее будет использовано общее обозначение “системы автоматизации экспериментов” (САЭ).

Ввиду отсутствия в данной области общепризнанных стандартов, коллективы разработчиков зачастую дублируют работу друг друга. В работе [Л1] выполнен анализ 200 САЭ по результатам опроса 40 различных организаций АН СССР и союзных республик, представляющих практически все направления, нуждающиеся в автоматизации научных исследований. “На разработку и доведение каждой из 200 созданных систем автоматизации при наличии базовых средств затрачено в лучшем случае год, а в большинстве случаев до двух-трех лет. Для каждой системы программное обеспечение создается практически независимо” [Л1]. Со времени публикации работы [Л1] прошло более 20 лет. Выполненный автором анализ материалов конференций (например, NOBUGS с 2002 г.) и отчетов разработчиков технологических систем разработки САЭ (SCADA-систем) показывает, что за эти годы достигнут существенный прогресс в области методики разработки ПО САЭ, однако, как и раньше, сроки разработки не удовлетворяют пользователей. В связи с этим, безусловно, актуальны работы, направленные на обеспечение преемственности разработанного кода и на сокращение сроков разработки САЭ.

Одной из очевидных причин удлинения сроков разработки САЭ является комплексный характер этой работы, в частности то, что мы вынуждены использовать элементы программирования на этапе адаптации САЭ к новому методу исследования. Логично было бы ввести в САЭ инструменты для адаптации ее к различным методикам эксперимента и составу оборудования без привлечения программирования. В системах типа SCADA появилась тенденция частичного отказа от использования традиционных языков программирования, однако при этом вводятся средства графического программирования, также предполагающие привлечение профессионалов [Л2].

Наибольшее влияние на преемственность кода и сроки модификации САЭ оказывают способ учета методики эксперимента и необходимость расширения состава специального оборудования. В **Таблице 1** САЭ, разработанные с помощью традиционных средств, условно разделены на две основные группы. **Интерпретирующие** – это системы, имеющие в своем составе интерпретатор. **С фиксированным** набором управляющих параметров - это системы с “жесткой” логикой, имеющие ограниченный набор предусмотренных методик и фиксированный интерфейс пользователя.

Выделим две проблемы, которые существенным образом влияют на решение задачи сокращения сроков модификации САЭ.

**1. Адаптация системы к прикладной методике.** Пользователи интерпретирующих систем могут в широком диапазоне изменять прикладную

методику, однако, независимо от состава изменений, требуется модификация управляющего скрипта, что негативно сказывается как на сроках выполнения этой задачи, так и на преемственности системы.

“Фиксированные” системы, в зависимости от реализации, либо вообще не предусматривают изменение прикладной методики, либо содержат ограниченный набор предусмотренных в ней методик. В случае, если требуемая прикладная методика выходит за рамки предложенного набора, необходимо выполнить работу программирования, компиляции и сборки системы. Учитывая то обстоятельство, что эта работа не предусматривается для выполнения ее пользователями, а выполняется разработчиками, кроме того, интерфейс полученной системы будет изменен, можно приравнять такую работу к разработке новой САЭ.

Таким образом, проблема состоит в том, что традиционные способы разработки при изменении методики эксперимента требуют использования программирования.

2. Второй аспект - **изменение состава оборудования**. Часто изменение методики приводит к изменению состава управляющих параметров системы, и как следствие – изменению состава оборудования, участвующего в эксперименте. В интерпретирующих системах эта проблема решается с помощью модификации управляющего скрипта, т.е. программирования. В системах с фиксированным составом методов исследования проблема не может быть решена без привлечения разработчиков. При этом разработчик будет вынужден изменить исходный код и интерфейс пользователя, заново выполнить трансляцию и сборку системы, вследствие чего, полученную систему можно будет назвать новой разработкой, новой системой.

Проблема состоит в том, что мы хотим избежать использования и программирования, и использования среды программирования, чтобы процедуру модификации ПО САЭ мог выполнять пользователь.

**Таблица 1.** Традиционные способы модификации САЭ

<b>Тип \ Цель</b>	<b>Изменение прикладной методики эксперимента</b>	<b>Изменение состава оборудования</b>
<b>Интерпретирующие системы</b>	Скрипт, программирование	Скрипт, программирование
<b>Фиксированные системы</b>	Ограниченный набор предусмотренных методик	Программирование и перекомпиляция системы

До последнего времени считалось, что разработка и модификация САЭ без применения программирования невозможна по причине непредсказуемости возможного развития метода выполнения эксперимента. Анализ требований к САЭ, условий использования и средств для их разработки в рассматриваемой области, выполненный в главе 1, позволил сделать вывод, что в настоящее время в достаточной мере развита методическая база, чтобы поставить в диссертации задачу разработки программного обеспечения такой системы и ее применения для автоматизации экспериментов в области физики низких энергий.

**ОСНОВНАЯ ЗАДАЧА:** разработать ПО и среду, позволяющую **без перетрансляции** компоновать из готовых модулей программную САЭ, код которой в рамках определенной области приложений **инвариантен относительно изменений метода исследования и конфигурации** оборудования.

В процессе работы автором найдено компромиссное решение обозначенных выше проблем.

**Решение проблемы адаптации системы к прикладной методике** заключается в фиксировании алгоритма выполнения методики. Это стало возможным благодаря использованию следующих методов:

1. Представление подсистемы управления в виде конечного автомата с повторным входом (reentrant), базирующегося на понятии "состояние". Состояния нумеруются, номер реализованного состояния запоминается и используется для инициализации системы при повторном входе. В каждое конечное состояние вложено одно и то же действие – экспозиция регистрирующей системы.
2. **Метод параметрического управления** (возможность обеспечена предыдущим методом, не требуется использования явного вызова процедур при составлении задания).
3. Представление задания в виде списка состояний САЭ.
4. Использование названий в терминах предметной области.
5. Инкапсуляция специфичных для аппаратуры алгоритмов перехода устройств из одного состояния в другое внутри драйверного слоя.

**Решение проблемы адаптации системы к составу оборудования** заключается в разработке драйверного слоя, используя методы:

1. **Представление драйвера в виде конечного автомата**, базирующегося на понятии "состояние". В качестве входных символов приняты названия параметров и их значения. При пуске САЭ драйвер получает набор начальных значений полного списка его параметров, в дальнейшем – "усеченный". При этом обеспечивается единообразное управление всеми объектами.
2. Драйвер для каждого устройства – отдельный, функционально законченный модуль в исполняемом формате.

3. Используется автоматическая регистрация драйверов в системе (подключение к исполняющим программам).
4. Используется единый способ композиции, а также единый интерфейс системы к драйверу (выполнена унификация).

Для реализации этих методов разработаны соответствующие структуры, алгоритмы и программы.

**МЕТОДЫ исследования.** При решении поставленных задач использовались методы математического и физического моделирования, абстрагирования, системного анализа, методы объектно-ориентированного программирования и разработки интеллектуальных систем, Интернет технологии и др. Автором в рамках созданного ПО разработаны и использованы новые методы: метод представления подсистемы управления состоянием САЭ и драйверов в виде конечных автоматов, метод параметрического управления состоянием САЭ, метод составления задания исполняющей системе в виде списка состояний системы, и др.

**ЗАЩИЩАЕМЫЕ ПОЛОЖЕНИЯ.** В рамках созданного ПО систем автоматизации экспериментов **разработаны новые технологии**, в том числе:

1. **Представление подсистемы управления состоянием САЭ в виде конечного автомата** [8]. Данная концепция позволила разработать метод параметрического управления состоянием САЭ и унифицировать способ управления САЭ и драйверами.
2. **Метод параметрического управления САЭ**, при котором пользователь описывает только условия работы, задавая названия и значения параметров, определяющих эти условия [8]. Данный метод, в сочетании со специальной структурой компонентов драйверного слоя программ [5], позволил создать **универсальную** программу подготовки задания на эксперимент, адаптирующуюся к используемой конфигурации оборудования САЭ. Основные достоинства метода:
  - исключается необходимость изучения специального языка, описывающего методику решения задачи;
  - доступны все возможные (в данной конфигурации) варианты методов исследования, и в то же время интерфейс пользователя не является избыточным.
3. **Метод составления задания исполняющей системе в виде списка состояний конечного автомата** и средства настройки интерфейса пользователя на заданную пользователем методику эксперимента [8].
4. Специальная **структура компонентов драйверного слоя** программ, обеспечивающая возможность автоматизировать включение их в состав программной системы - подключение к исполняющей подсистеме и к подсистеме составления задания [5,6,7]. Эти компоненты также представляются в виде конечных автоматов.

5. **Специальная структура САЭ**, включающая унифицированные базовые модули и драйверный слой программ, а также способ интеграции модулей в систему [5,9]. Достоинства принятого решения следующие:
  - существенная экономия времени разработчиков благодаря использованию значительного объема проверенного кода базовых модулей и преемственности компонентов драйверного слоя;
  - возможность масштабирования САЭ без перетрансляции;
  - при наличии готовых модулей - возможность выполнить модификацию системы силами пользователя, включая и расширение конфигурации САЭ.
6. **Унифицированная подсистема передачи данных**, предназначенная для использования в распределенных САЭ со смешанным составом аппаратных средств коммуникации [9].

**НАУЧНАЯ НОВИЗНА** работы определяется тем, что в ней при **разработке программного обеспечения** систем автоматизации экспериментов автором решены актуальные задачи, имеющие самостоятельное научное и практическое значение:

1. Разработан и **впервые применен метод представления драйверов и подсистемы управления состоянием САЭ в виде конечных автоматов**. Данный метод лег в основу метода параметрического управления состоянием САЭ и позволил унифицировать способ управления САЭ и драйверами.
2. Разработан и **впервые применен метод параметрического управления состоянием САЭ**. Такое управление позволило создать **универсальную** программу подготовки задания на эксперимент в виде описания конечного автомата (списка состояний системы).
3. **Впервые** поставлена и решена задача разработки программной среды, позволяющей **без перетрансляции** компоновать из готовых модулей САЭ, код которой **остается неизменным при модификации** методики исследования и конфигурации оборудования. Реализация этой концепции позволяет существенно сократить время разработки и развития программного обеспечения САЭ.
4. Разработана **оригинальная структура ПО САЭ**, основанная на использовании **нового метода** параметрического управления ее состоянием, позволяющая интегрировать САЭ из готовых модулей в загрузочном формате силами пользователей.

Основные положения, выводы и рекомендации доведены до практической реализации и выдержали проверку в эксплуатации. Разработанные при участии автора программные системы позволили получить важные **научные результаты** в экспериментах на реакторе ИБР-2 [Л4, Л5].

**ПРАКТИЧЕСКАЯ ценность.** Работа выполнялась в соответствии с проблемно-тематическими планами НПЦ "АСПЕКТ" и протоколами о сотрудничестве с ЛНФ ОИЯИ. Результаты данной работы нашли практическое применение в системах автоматизации экспериментов на реакторе ИБР-2.



Помимо этого, в Приложении-1 к диссертации приведены названия 9 массовых потребителей систем контроля транспортных средств и грузов, систем радиационного контроля и др., разработанных с участием диссертанта. В Приложении-2 даны названия 10 программных продуктов, используемых с 1997 года на таможенных станциях, ядерных объектах России, ближнего и дальнего зарубежья. Тираж этих продуктов – до 300 шт. Некоторые из доложенных методов и алгоритмов проверены в процессе эксплуатации этих систем. Проблемы, с которыми сталкивался автор, опыт, приобретенный при разработке и внедрении этих приложений, а также анализ результатов эксплуатации дали важную информацию для выполнения данной работы. Приложение-3 содержит отзывы потребителей. Опыт работы автора в НПЦ “АСПЕКТ” показывает, что разработка, особенно – сопровождение этих систем малым коллективом, значительно упрощается при использовании предложенных методов.

**АПРОБАЦИЯ работы.** Отдельные разделы работы были представлены на Всероссийской научно-технической конференции “Методы и средства обработки информации (МСО-2005)”, 5-7 окт. 2005, МГУ; XVIII Совещании по использованию рассеяния нейтронов в исследованиях конденсированного состояния. 12-16 октября 2004, Заречный, Россия; 57 Международной конференции “Ядро 2007” “Фундаментальные проблемы ядерной, атомной энергии и ядерных технологий”, 25-29 июня 2007, Воронеж. Основные результаты работы докладывались на научных семинарах ЛНФ ОИЯИ, ЛИТ ОИЯИ, НИИПА, АСПЕКТ, ВМК МГУ, ФЭИ. Ряд работ представлен в отчетах по гранту РФФИ № 04-07-90256в.

**ПУБЛИКАЦИИ и личный вклад автора.** По результатам диссертации опубликовано 9 работ [1..9]. В работе [1] автору принадлежит разработка алгоритмов и программного обеспечения системы. Аппаратная часть системы [1] и программы окончательной математической обработки разработаны соавторами и не описываются в диссертации. Вопрос о возможности разработки САЭ, инвариантной относительно изменений методики ее использования, впервые был поднят в работе [Л6]. Однако в этой работе допускалось изменение методики только в рамках фиксированной конфигурации оборудования, что является частным случаем и не соответствует сегодняшним запросам. В диссертации, в отличие от [Л6], сформулирована концепция системы, инвариантной относительно изменения **конфигурации оборудования САЭ**, отражающей прикладную методику [5, 9]. Автором разработаны основные алгоритмы и методы, выполнена проверка реализуемости предложенной концепции на практике. Результаты, выносимые на защиту, получены лично автором. В работах, опубликованных в соавторстве, личный вклад автора был определяющим на всех этапах выполнения данных работ.

**Структура и объем диссертации.** Диссертационная работа состоит из введения, 6 глав и заключения. Она изложена на 113 страницах машинописного текста, содержит 26 рисунков, 3 таблицы, 3 приложения. Список литературы включает 93 наименования.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** сформулирована постановка задачи и обосновывается ее актуальность. Приведен список требований к системам автоматизированного управления, проблем, ожидаемых в процессе решения поставленной задачи, и предлагаемые пути их решения.

**В первой главе** приведен обзор методов разработки ПО для систем автоматизации экспериментов и обоснование выбора ОС Windows в качестве среды для данной разработки. Состав методов для представления в обзоре выбран на основании анализа материалов конференции NOBUGS с 2002 г. Рассмотрены свойства прагматического подхода к задаче, использование интерпретаторов, специальных систем (TACO, EPICS), средств RAD (Delphy) и систем типа SCADA (LabView и др.). Приведены примеры технологичного подхода (OPC, CAN) к разработке драйверного слоя САЭ. Материалы обзора показывают, что

- возникло и приобретает все больший вес использование технологических систем разработки САЭ;
- наметилась тенденция к отказу от использования традиционных языков программирования при решении задачи интеграции САЭ из программных компонентов, в известных системах (в основном - SCADA) это проявляется в использовании графических средств программирования;
- для решения задачи разработки программного обеспечения в исследовательских организациях создаются коалиции с коллективной разработкой и использованием продукта.

На основании рассмотренных материалов автор пришел к следующим **выводам**:

1. Использование технологической инструментальной среды программирования из числа известных не обеспечивает малые сроки разработки ПО САЭ, отчасти из-за необходимости взаимодействия коллективов с различной специализацией. Сокращение сроков разработки, отмечаемое авторами той или иной технологии, проявляется при попытке разработать эквивалентную по сложности САЭ другим методом. Зачастую без использования нужной технологии эквивалентную систему построить просто нереально. Главный выигрыш от использования технологической среды разработки – это возможность при тиражировании подобных систем получать более высокий уровень автоматизации, сервиса и надежности САЭ за счет использования многократно проверенного кода ряда компонентов и автоматизированной сборки системы.

2. Представляет интерес разработка средств, позволяющих при изменении способа использования САЭ (т.е. метода исследования) сохранить преемственность ранее разработанного кода САЭ.

3. Программы, непосредственно работающие с оборудованием, составляют менее 10% общего объема САЭ, однако занимают львиную долю во времени разработки. В связи с этим актуальны разработки, обеспечивающие

преимущество разработанного кода драйверов, унификацию способа их использования и объединение среды моделирования с рабочей средой.

4. Подготовка к проведению эксперимента в исследовательской организации часто занимает намного больше времени, чем его проведение. Одна из причин этих потерь - в необходимости привлекать специалистов различного профиля, в частности - программистов. Этот факт вполне осознан разработчиками инструментальных систем. Ряд современных SCADA-систем уже позволяют выполнить интеграцию САЭ без использования традиционных языков программирования [Л12]. Однако при этом вводятся специальные системы графического программирования, что опять подразумевает привлечение квалифицированных специалистов, владеющих методикой работы с данной SCADA.

Разработка специального инструмента для модификации программного обеспечения САЭ и передача его в руки самих пользователей существенно сократит потери времени. Разработка ориентированного на определенную прикладную область графического редактора является перспективным решением, при условии, что такой редактор также может быть настроен оператором и для его использования не потребуются специальные знания в области программирования.

5. Характеристики ОС Windows достаточны для реализации интересующих нас САЭ.

Анализ материалов NOBUGS (130 статей) показывает, что известные САЭ можно отнести к двум группам, характеристики которых представлены в **таблице 1**. Характерной особенностью всех рассмотренных средств является то, что для разработки новой или модернизации существующей САЭ требуется определенный объем программирования. Результатом этого может стать увеличение сроков модернизации, которое может быть неадекватно увеличению сложности самой системы. Предлагаемый подход позволяет получить ту же универсальность, что и использование интерпретатора, но дает преимущество благодаря устранению необходимости использовать программирование.

Таким образом, поставленная в диссертации актуальная задача сформулирована на основании анализа состояния и свойств современных средств разработки САЭ и направлена на обеспечение надежности кода САЭ, преимущественности программ и сокращения сроков разработки.

Во **второй главе** сформулирована концепция программы управления системой автоматизации, использующей представление задания в виде описания конечного автомата, заданного списком состояний, и предложена специальная структура САЭ.

С точки зрения специалиста по информатике, автоматизируемый объект состоит из следующих компонентов:

- источников данных (детекторов, датчиков, каналов передачи данных, ...);
- памяти, в которой буферизуются регистрируемые данные;
- устройств, изменяющих состояние объекта или (что то же) условия его работы (терморегуляторы, двигатели, заслонки, коллиматоры и др.).

Такое оборудование функционально может управляться набором программ драйверного уровня, выполняющих следующие операции:

- инициализацию оборудования, обеспечивающую его работоспособность;
- управление состоянием оборудования – занесение (запись) значений, определяющих требуемое (на данный момент) методикой эксперимента состояние всей САЭ.

Рассмотрим **возможный способ реализации методики исследования на этапе регистрации данных.**

Выделим программы, работающие непосредственно с оборудованием автоматизируемого объекта, в отдельный слой программ (драйверный слой). Введем специальную программу (назовем ее программой управления состоянием), которая, в сочетании с драйверным слоем и объектом, реализует **концепцию конечного автомата (КА)**. В соответствии с такой концепцией, программе управления состоянием может быть передано описание конечного состояния объекта, и после перехода системы в новое состояние об этом может быть получен сигнал. Таким способом мы можем на уровне обращения к программе управления состоянием перейти от функционального способа управления (вызова функциональных программ) к параметрическому.

Структуру программного обеспечения САЭ представим в виде нескольких универсальных базовых программ. Для дальнейшего рассмотрения введем понятие параметрической модели объекта, как списка параметров, характерных для текущей конфигурации и определяющих возможные методики исследования. Это могут быть, например, температура, номер мишени, положение устройства, порог аварийной сигнализации, продолжительность экспозиции данных и т.п.

Основа обсуждаемой структуры - это четыре программы, каждая из которых не зависит от конкретной прикладной методики:

1. **Программа диалога с пользователем**, в котором он может определить набор изменяемых параметров для данного сеанса работы, являющийся подмножеством параметрической модели объекта, задать последовательность значений этих параметров (для каждого параметра - свой вектор значений) и очередность их изменения. Мы получим матрицу значений изменяемых параметров, в которой каждому используемому параметру отведен столбец, а порядок столбцов определяет очередность изменения значений. В данной разработке существенно то, что мы представляем их в виде списка строк, содержащего только названия параметров и нужные значения. Такой набор данных (описание прикладной методики для конкретного сеанса работы) является **табличным представлением автомата**, строка матрицы описывает его конечное состояние при выполнении очередного измерения. Данная диалоговая программа в качестве основной входит в состав **подсистемы подготовки задания**.
2. **Программа управления работой САЭ**, выбирающая из заданной матрицы строку и передающая ее исполняющей подсистеме для выполнения очередного измерения.

3. **Программа управления состоянием** устанавливает условия измерения, используя для этой цели заданное описание **конечного состояния** и готовые программы драйверного уровня ПО установки. Данная программа является основным модулем в исполняющей подсистеме САЭ.
4. **Подсистема предварительной обработки** данных выполняет ряд системных функций общего назначения, а также вызывает функции обработки данных, состав и последовательность выполнения которых определяется областью приложения. Важной системной функцией является обратная связь между подсистемами регистрации и обработки данных. Помимо этого, к функциям общего назначения относится регистрация факта завершения работы с текущим вектором состояния, что обеспечивает возможность возобновления работы при аварийных сбоях операционной системы, системы питания, сброса мощности и др. при минимальных (прогнозируемых) потерях накопленной к этому моменту информации и др.

Построение такой привязки к методике исследования и составу оборудования позволяет на различных объектах использовать одну и ту же группу унифицированных базовых программ, которые будут работать следующему алгоритму:

- выбрать из списка состояний (задания) строку - описание состояния КА;
- установить на объекте условия в соответствии с описанием состояния;
- выполнить экспозицию – выполнить измерения и зарегистрировать данные;
- отправить данные в архив или на обработку, зарегистрировать факт завершения работы с текущим вектором состояния;
- повторять предыдущие шаги, пока не исчерпается список состояний.

Возможность реализовать предложенную концепцию не очевидна. Поэтому автором выполнена разработка системы, реализующей сформулированные принципы, с целью проверить на практике прогнозируемые характеристики. Ввиду сложности задачи ее решение разбито на этапы. Первая реализация упрощена и предназначена для проверки возможности **использовать программы без изменений в различных САЭ**. Разработка включает программу **подготовки задания** и необходимые унифицированные модули: программу управления работой САЭ, программу управления состоянием САЭ и подсистему предварительной обработки (рис. 1).

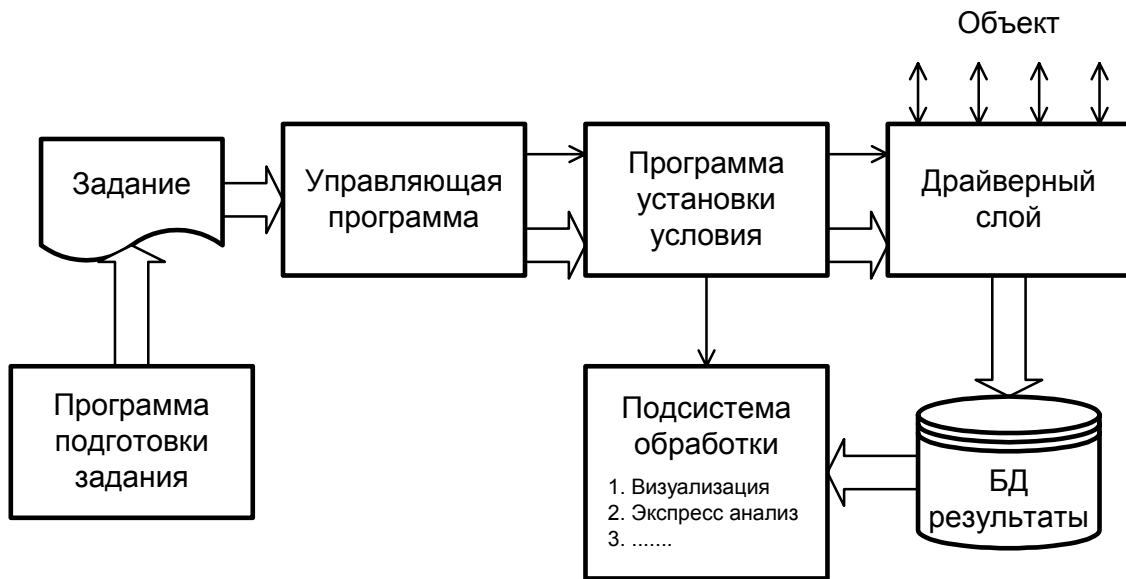


Рис. 1. Упрощенная блок-схема системы автоматизации эксперимента

Программа управления работой САЭ **PULT** (см. рис. 2) обеспечивает выполнение задания, представленного в виде списка состояний. Эта же программа в процессе выполнения задания исполняет команды пользователя, введенные с клавиатуры или принятые по сети. В ее подчинении находится ряд унифицированных модулей и подсистем, состав которых может меняться в зависимости от запросов пользователя и по мере развития данного комплекса, однако основная группа модулей фиксирована и выполняет следующие функции:

- выбирает из матрицы задания строку очередного состояния;
- устанавливает очередное состояние САЭ;
- вызывает подсистему обработки данных; решает вопрос о возможности зарегистрировать данные, как успешно измеренные, и перейти к измерению следующего файла, повторению предыдущего либо к завершению задания.

Помимо этих основных модулей, в цепочку исполнения программой **PULT** могут включаться модули визуализации данных и состояния САЭ, представления данных в Интернет, математической обработки данных в реальном времени, программы пользователя и др.

Взаимодействие между управляющей программой и подчиненными модулями осуществляется посредством команд и ответных сообщений, обслуживаемых специальной подсистемой.

Подсистема **обработки данных** выполняет:

1. архивирование данных;
2. графическое представление данных;
3. контроль и протоколирование условий измерения каждого файла;

4. представление в реальном времени в Интернет динамически формируемой информации о состоянии спектрометра, экспериментальных данных и программ;
5. вызов программ пользователей, выполняющих нестандартные операции обработки.

Первые 4 функции подсистемы обеспечиваются программами, не зависящими от конкретной задачи. Подключение программ пользователей выполняется списком. Программы пользователя могут быть написаны на любом языке. Возможно подключение модулей, разработанных в среде LabView, MATLAB и др.



Рис. 2. Интерфейс пользователя управляющей программы в режиме выполнения задания

Базовое программное обеспечение, построенное в соответствии с изложенными принципами, **пригодно для использования без изменений в различных САЭ** и дает **сокращение объема необходимого программирования** при одновременном увеличении реализуемых возможностей. Без дополнительного программирования реализуется возможность получить

- средства комплексной проверки и тестирования САЭ при любых сочетаниях варьируемых условий работы;
- подробную информации об условиях измерения и структуре любого файла данных, визуализацию состояния данных и программ, в том числе средствами Интернет.

Фрагменты описанной системы являются развитием алгоритмов, которые успешно использовались в течение ряда лет в ПО ряда спектрометров на реакторе ИБР-2 и системах, разработанных в НПЦ “Аспект”. Некоторые из предлагаемых решений могут найти и более широкое применение.

Автор считает, что предложенные алгоритмы еще далеки от совершенства, продолжают развиваться, однако уже **данная разработка, основанная на представлении САЭ в виде конечного автомата, подтвердила возможность реализации программного обеспечения САЭ, инвариантного относительно изменений методики исследования.**

Опираясь на полученное подтверждение основного из прогнозируемых свойств развиваемого подхода, автор выполнил разработку комплекса программ в виде масштабируемой распределенной системы.

**Третья глава.** В структуре программного обеспечения САЭ разделены средства подготовки задания и его исполнения - мы выделили автономную подсистему подготовки задания, которая должна автоматически адаптироваться к задаваемой конфигурации специального оборудования САЭ. В составе подсистемы подготовки задания присутствует база данных САЭ, обеспечивающая **информационную поддержку САЭ** при настройке ее модулей служебными программами, и программа подготовки задания **PSJ (Preparation of Single Job)**. В **базе данных** введены таблицы:

- **описания готовых компонентов** драйверного слоя, доступных разработчикам в данной организации – “библиотека” устройств;
- **таблица изменяемых параметров** драйверов, их типы, диапазоны значений и значения, принимаемые по умолчанию;
- **описание конфигурации** драйверного слоя разрабатываемой САЭ;
- **таблица дополнительных параметров**, используемых в программах обработки данных, при подготовке описаний данных, отчетов и др.;
- **список заданных программистом альтернативных названий параметров**;
- **некоторые другие таблицы.**

Таблицы изменяемых и дополнительных параметров составляют параметрическую модель управляемого объекта.

Программа **PSJ** представляет собой специальный редактор, который может работать в нескольких режимах. Основной режим - составление задания с новым списком изменяемых параметров. По команде открывается окно с тремя полями списков (рис. 3):

- **названия всех параметров спектрометра** (параметрическая модель, автоматически составленная на основании данных из БД);
- **параметры, изменяемые в планируемом сеансе работы** (составляет пользователь);
- **поле списков значений изменяемых параметров** (задает пользователь).

Данное окно предоставляет возможность выбрать названия тех параметров, значения которых будут изменяться в данном сеансе; задать нужную очередность изменения значений параметров; для каждого параметра задать список используемых в сеансе значений. После этого **PSJ** автоматически создает список состояний – табличное представление используемого конечного автомата. Пример такого списка представлен на рис. 4.

Программа **PSJ** использует понятия параметрической модели управляемого объекта и вектора состояния системы, настраивается в соответствии с документацией САЭ, хранящейся в БД, инвариантна относительно изменений методики исследования и конфигурации САЭ и пригодна для **использования без изменений в различных САЭ.**

Использование рассмотренного способа подготовки и исполнения задания в виде описания конечного автомата **сокращает объем необходимого программирования** при одновременном увеличении реализуемых возможностей.



Без дополнительного программирования, благодаря использованию редактора PSJ и концепции КА, реализуется:

- автоматический и диалоговый режимы работы;
- возможность вмешаться в автоматическую работу САЭ для коррекции задания в процессе его исполнения;
- автоматическое определение точки входа (номера состояния) для продолжения работы САЭ после сбоя питания и др. с известными и восстанавливаемыми потерями данных, рестарт системы без участия пользователя.

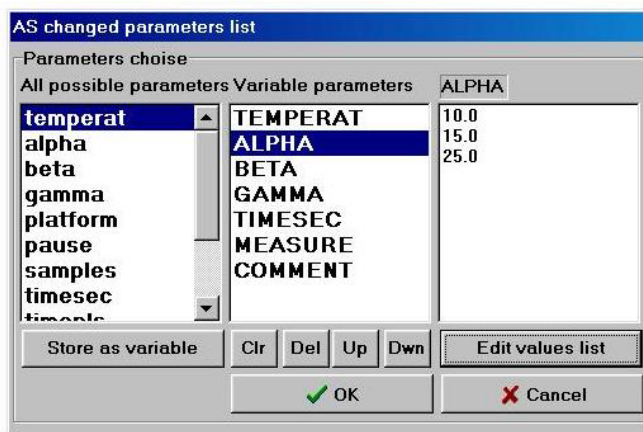


Рис. 3. Окно редактирования списка изменяемых параметров и списков их значений

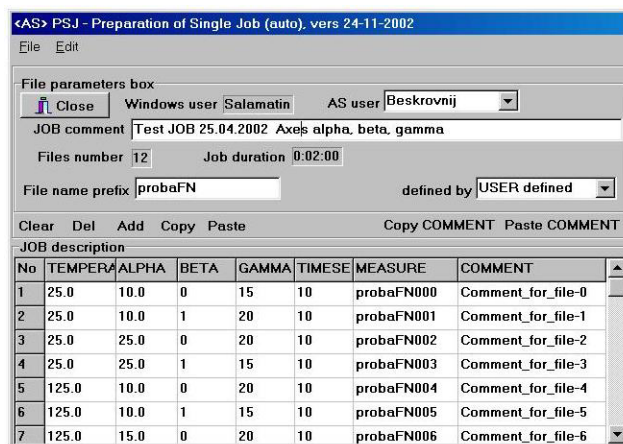


Рис. 4. Задание (список состояний КА), составленное программой PSJ

В **четвертой** главе рассмотрены вопросы разработки САЭ в виде распределенной системы, масштабируемой силами пользователя.

Для развития среды разработки и расширения сервиса использованы 3 варианта способов взаимодействия компонентов САЭ:

- по протоколу TCP/IP, обеспечивающему аппаратную независимость;

- “программная общая шина”, образуемая через “файл, отображенный в памяти” (возможность реализована в ОС);
- Web-интерфейс.

Базовые модули и драйверы могут помещаться на разных ЭВМ и используют взаимодействие по протоколу TCP/IP.

Подсистемы оперативной визуализации на пульте оператора, предварительной обработки и др. используют интерфейс “программная общая шина”.

Для удаленного управления, визуализации состояния САЭ и данных используется Web-интерфейс.

В интересующих нас системах автоматизации процессор, как правило, слабо загружен (< 10%). Поэтому масштабируемость нас интересует, как способность автоматически адаптироваться к изменяемому составу экспериментального оборудования и количеству ЭВМ, которые САЭ использует в процессе работы. Для реализации этих свойств потребовалось разработать специальную структуру драйверов, развить логическую структуру САЭ, расширить состав базовых модулей комплекса.

При разработке **структуры драйвера** преследовались цели:

- создание САЭ в виде распределенной масштабируемой системы;
- обеспечение преемственности и минимизация кода драйвера;
- унификация способа вызова к драйвера;
- автоматизация интеграции драйверов в систему – подключения к подсистемам подготовки и исполнения задания.

Для обеспечения единообразного управления всеми объектами драйверы представлены в виде **конечных автоматов**. Входными символами являются **названия параметров и их значения**, а в таблицу переходов помещены **адреса соответствующих процедур**.

Драйверы данного комплекса включают интерфейсную часть и функциональный модуль. Назначение **интерфейсной** части - выполнение ряда служебных функций без обращения к функциональному модулю:

- предустановка значений параметров устройства;
- прием команды от программы **SetVec**;
- выполнение рутинных преобразований команды в описание нужного состояния драйвера (устройства) и вызов требуемой прикладной функции драйвера;
- предоставление драйверу процедуры асинхронного вызова клиента для передачи информации о результатах выполнения команды.

**Функциональный** модуль включает стандартный набор функций:

- **INIT** - включение устройства, инициализация программы и устройства;
- **OFF** – выключение электропитания устройства;
- **READ** - чтение из устройства текущего значения параметра;
- **SET** - выполнение действия, ассоциированного с параметром (“установить условия”).

Задача разработчика драйвера сведена, в основном, к кодированию способа выполнения нескольких специальных функций, ради которых данное устройство включено в состав установки, и объединения их с интерфейсным модулем. Такая структура драйвера:

- 1) экономит время разработчика;
- 2) обеспечивает преемственность функциональных модулей, свободных от наложения кода, отражающего прикладную методику;
- 3) повышает надежность комплекса программ благодаря использованию многократно проверенно кода интерфейса.

**Структура распределенной САЭ** представлена на рис. 5. По функциональному назначению в составе ПО САЭ мы выделяем: **ЭВМ Пульта оператора, Сервер приложения** и **ЭВМ-Контроллеры**. Временно подключающиеся **ЭВМ Наблюдателей** используют штатный Internet-браузер. Возможное назначение ЭВМ определяется составом загруженных в неё базовых программ, как это показано на рис. 5. Для реализации данной структуры разработаны базовые модули **Terminal, TCPclient, TCPserver, SetVec, FrontEnd**. Функции логических ЭВМ данного комплекса описываются ниже.

На **Пульте оператора** работает программа **Terminal** - интерфейс пользователя, позволяющий управлять параметрами и режимом работы САЭ.

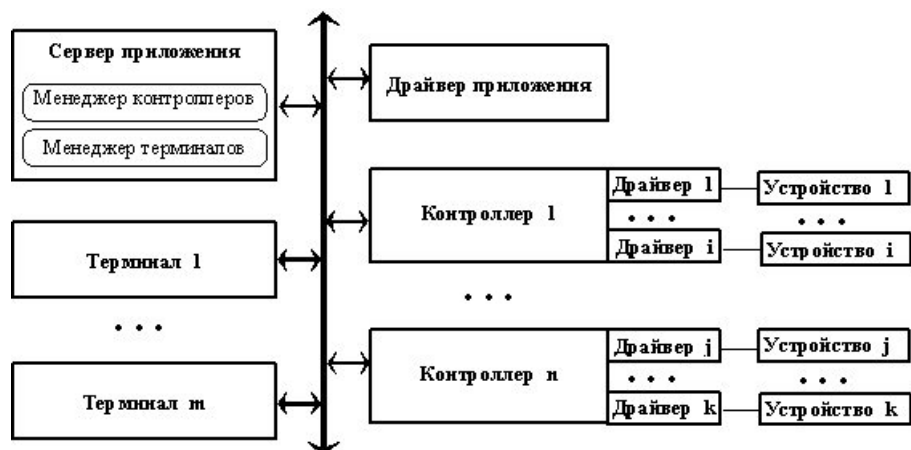


Рис. 5. Логическая схема САЭ

**Сервер приложения** осуществляет:

- управление конфигурацией САЭ – регистрирует подключающиеся Пульты операторов и Контроллеры;
- управление конфигурацией Контроллеров - составом драйверов на основании данных из документации САЭ в БД;
- выполнение задания, представленного списком состояний **КА (SetVec)**;
- передачу команд драйверам и прием ответных сообщений (**TCPserver**).

Основная функция - выполнение задания – осуществляется модулем **SetVec**. Модуль выбирает из списка очередную строку состояния КА. На основании данных о предыдущем и требуемом состояниях программа **SetVec**

выполняет следующие действия:

1. вычисляет вектор изменения состояния САЭ – формирует сокращенное описание состояния КА;
2. составляет список команд управления отдельными узлами оборудования САЭ;
3. оптимизирует последовательность передачи команд конкретным драйверам;
4. передает драйверам группу команд; принимает асинхронные отчеты драйверов о выполнении команд;
5. повторяет пункты 3,4, пока не достигнет заданного состояния объекта.

Работа программы **SetVec** считается завершенной, если исчерпан список состояний и получены отчеты от всех драйверов, либо работа прервана по аварии (по таймауту).

Основная функция **ЭВМ-Контроллера** - управление подключенным к ней оборудованием спектрометра. Ее программное обеспечение включает базовые программы **TCPclient**, **FrontEnd** и драйверы. Программа **TCPclient** принимает команды, поступающие от управляющей ЭВМ (от программы **SetVec**), и передает асинхронные сообщения от драйверов на **Сервер приложения**. Программа **FrontEnd** выполняет диспетчеризацию команд – передает их соответствующим драйверам.

Описанные программы позволяют выполнять работу САЭ полностью в автоматическом режиме. При отказах САЭ в этом режиме потери могут быть невосполнимы, поэтому целесообразна проверка работоспособности САЭ при различных составах и последовательности работы узлов оборудования. Разработка полного набора тестов - задача неперспективная из-за большой трудоемкости и невозможности предусмотреть все варианты конфигурации и развития физических методик. Вместо этого была разработана специальная программа **МС** (Manual Control) для интерактивного управления работой САЭ [3]. При разработке **МС** были поставлены задачи:

- дать инструмент для быстрой идентификации и замены сбойного оборудования или программы;
- дать пользователю, в дополнение к автоматическому режиму, возможность выполнять простые работы в режиме интерактивного управления установкой.

Такой инструмент полезен для работы с любой системой, поэтому были приняты меры для обеспечения автономности программы **МС** путем дублирования или упрощения некоторых функций, реализованных в подсистеме подготовки задания. Разработанная программа **МС** позволяет в пределах заданной параметрической модели установки выполнить проверку реализуемости любых состояний спектрометра, определяемых доступным оборудованием и заданным диапазоном допустимых значений (любые сочетания параметров и последовательность выполнения операций управления).

Описанные в данной главе методы и алгоритмы реализованы в базовых программах данного комплекса [5] и использованы в программном обеспечении спектрометров **COCOS**, **DN-2** [2-4, Л3, Л4] на реакторе ИБР-2 и беспроводной системы [1]. Эксплуатация САЭ для спектрометра **DN-2** [5, 6, Л4] и системы [1] в

течение 2003-2006 г. подтверждает возможность средствами данного комплекса:

- реализовать распределенную систему автоматизации экспериментов;
- легко масштабировать систему автоматизации (изменять количество ЭВМ и драйверов) силами пользователя;
- использовать ЭВМ-контроллеры, работающие под любой операционной системой, допускающей соединение по протоколу TCP/IP;
- использовать в САЭ оборудование различных стандартов одновременно.

**В пятой главе** описана разработка унифицированной подсистемы передачи данных (ППД) в САЭ со смешанным составом аппаратных средств коммуникации [9]. В общем случае в разработанной структуре САЭ можно выделить два уровня коммуникации: уровень управляющих ЭВМ и драйверный уровень. На основе анализа и исследования свойств известных механизмов взаимодействия процессов в ОС Windows обоснован выбор протокола TCP/IP для уровня управляющих ЭВМ. Подробно рассмотрена организация ППД для драйверного уровня. Унификация данной подсистемы освобождает пользователя от рассмотрения сложных вопросов.

**В шестой главе** для демонстрации возможностей программного комплекса описаны некоторые приложения:

1. Распределенная беспроводная система регистрации импульсных физических полей с синхронизацией потоков данных [1]. Данный пример демонстрирует возможность динамического изменения конфигурации распределенной САЭ, возможность реализовать быстрые процессы (работа отдельных ЭВМ, не имеющих проводного соединения, может быть синхронизирована с погрешностью до 100 нс), показана возможность адаптировать интерфейс пользователя к запросам потребителя, не изменяя остального кода САЭ и др.
2. Программа **МС** диалогового управления работой САЭ [6]. Данная программа позволяет выполнить проверку реализуемости любых состояний спектрометра, определяемых доступным оборудованием и заданным диапазоном допустимых значений. Программа может без изменения работать с различными САЭ, разработанными по описанной методике.
3. Подсистема диалогового анализа данных в прецизионных экспериментах [7]. Данная подсистема разработана на основе модуля автоматического анализа данных. Модуль является примером программ, которые включаются в состав подсистемы обработки данных и в режиме On-Line могут вырабатывать сигналы обратной связи для управления работой САЭ.

## **ЗАКЛЮЧЕНИЕ**

**Основным результатом** диссертации является создание программного обеспечения САЭ на основе новых методов, в том числе:

1. Представление драйверов и исполняющей подсистемы САЭ **в виде конечных автоматов**. Данный метод лег в основу метода параметрического управления

- состоянием САЭ и позволил унифицировать способ управления САЭ и драйверами.
2. Разработка **метода параметрического управления САЭ**. Такое управление позволило создать **универсальную** программу подготовки задания на эксперимент в виде описания конечного автомата (списка состояний системы), адаптирующуюся к используемой конфигурации оборудования САЭ.
  3. Разработка **метода составления задания** исполняющей системе в виде **списка состояний конечного автомата** и средств настройки интерфейса пользователя на заданную пользователем методику эксперимента в рамках избыточной и изменяемой конфигурации системы
  4. Разработка специальной **структуры компонентов драйверного слоя** программ, обеспечивающей унификацию обращения к драйверам и возможность автоматического включения их в состав программной системы - подключения к исполняющей подсистеме и к подсистеме составления задания.
  5. Разработка **специальной структуры ПО САЭ**, основанной на использовании **нового метода** параметрического управления ее состоянием, включающей унифицированные базовые модули и драйверный слой программ, а также способа интеграции модулей в систему. Базовые модули и способ интеграции системы пригодны без изменения для решения широкого класса задач. Принятое решение позволило сократить объем программирования при одновременном расширении возможностей, в числе которых возможность выполнить интеграцию модулей в систему силами пользователей без перепрограммирования САЭ.
  6. Разработка **унифицированной подсистемы передачи данных**, предназначенной для использования в распределенных САЭ со смешанным составом аппаратных средств коммуникации.

Разработанные методы и программы позволили построить среду для интегрирования программных модулей в ПО САЭ без перетрансляции ее компонентов.

Описанные методы и программы использованы при разработке ряда систем для конкретных применений, часть которых перечислена в Приложениях 1, 2 к диссертации. При их использовании получены важные научные и практические результаты [Л5, Л6].

**Общим итогом работы** явилась разработка ПО САЭ и проверка на практике среды для интеграции готовых программных модулей в масштабируемую программную систему автоматизации экспериментов силами пользователя.

**Результаты публиковались** в научных журналах [6,7], рекомендованных ВАК, в виде сообщений ОИЯИ [1-3,8], докладывались на всесоюзных конференциях и совещаниях [4,5,7,8], а также представлены в отчетах за 2005 и 2006 г. по теме гранта РФФИ № 04-07-90256в.

**ПО ТЕМЕ ДИССЕРТАЦИИ  
ОПУБЛИКОВАНЫ СЛЕДУЮЩИЕ РАБОТЫ**

1. Н.В. Астахова, Л.Г. Бордюгов, А.В. Герасимов, Н.Д. Дикусар, Г.И. Еремин, А.И. Иванов, Ю.С. Крюков, Н.Г. Мазный и др. “Распределенная беспроводная система регистрации с синхронизацией потоков данных” //ОИЯИ Р13-2006-41, Дубна, 2006.
2. Н.В. Астахова, Н.Д. Дикусар, Н.Г. Мазный и др. “Программный комплекс АС (автоматизация спектрометрии). Управление окружением образца”//ОИЯИ Р13-2004-203, Дубна, 2004.
3. Н.В. Астахова, А.И. Бескровный, П.Е. Буторин, С.Е. Василовский, Н.Г. Мазный и др. “Комплекс АС (автоматизация спектрометрии). Интерактивное управление спектрометром”//ОИЯИ Р13-2004-204, Дубна, 2004.
4. Н.В. Астахова, Н.Д. Дикусар, Н.Г. Мазный и др. “Автоматизация спектрометрии - управление окружением образца” //Труды XVIII Совещания по использованию рассеяния нейтронов в исследованиях конденсированного состояния. 12-16 октября, Заречный, Россия. Программа и тезисы. ИФМ УрО РАН, Екатеринбург, 2004, с.147.
5. Н.Г. Мазный “Среда для быстрой разработки гибких распределенных систем автоматизации экспериментов” //Труды Всероссийской научно-технической конференции “Методы и средства обработки информации (МСО-2005)”, 5-7окт. 2005, МГУ; М: Изд. отд. фак. ВМиК МГУ, 2005, с.97-102.
6.
  - Н.В. Астахова, Н.Д. Дикусар, Н.Г. Мазный и др. “Программный комплекс АС (автоматизация спектрометрии). 3. Методика управления окружением образца и применение ее в программе интерактивного управления спектрометром” //ПТЭ, 2005, №5, с. 36-43.
  - N.V. Astakhova, N.D. Dikoussar, N.G. Maznyi et al. “An AS Software Complex for the Automation of Spectrometry, Part III: A Technique for Controlling the Sample Environment and Its Application in the Spectrometer Interactive Control Program”// **Instruments and Experimental Techniques, 2005, Vol. 48, No. 5, pp. 592-598.** М: МАИК "Наука/Interperiodica", 2005.
7. Н.А. Гундорин, Н.Д. Дикусар, Н.Г. Мазный и др. “Экспресс-анализ спектров в прецизионных экспериментах”
  - //Известия РАН, серия Физическая, 2009, том 73, №2, с. 261-263,
  - //Доклад на 57 Международной конференции “Ядро 2007” “Фундаментальные проблемы ядерной физики, атомной энергетики и ядерных технологий”, Воронеж, 25-29 июня 2007;
  - //ОИЯИ, Р10-2007-94, Дубна, 2007;
  - // **Bulletin of the Russian Academy of Sciences: Physics, 2009, vol. 73, №2, pp.249-251.**
8. Н.Г. Мазный, И.М. Саламатин, К.М. Саламатин. “Генерация программ автоматизации экспериментов из модулей в формате загрузки”
  - //ОИЯИ, Р13-2007-93, Дубна, 2007, 15с;

- //Доклад на 57 Международной конференции “Ядро 2007” “Фундаментальные проблемы ядерной физики, атомной энергетики и ядерных технологий”, Воронеж, 25-29 июня 2007.
9. Н.Г. Мазный. “Подсистема передачи данных между процессами в распределенной системе автоматизации” //ОИЯИ, Р10-2008-185, Дубна 2008.



### Литература.

- Л1. Е.П. Велихов, А.Н. Выставкин. “Проблемы развития работ по автоматизации научных исследований” // УСиМ, №4, 1984, стр. 3-12.
- Л2. О.С. Соболев. “Прогресс в области SCADA-систем и проблемы пользователей”// <http://www.mka.ru/?p=41527>.
- Л3. А.А.Шалыто, Н.И.Туккель. “Программирование с явным выделением состояний” // “Мир ПК”, 2001, №8, С.116-121; №9, С.132-138
- Л4. Н.А. Гундорин, К.В. Жданова, В.Е. Жучко и др. “Измерение выхода запаздывающих нейтронов при делении  $^{237}\text{Np}$  тепловыми нейтронами” // Ядерная физика, 2007, том 70, № 6, стр. 1-8.
- Л5. Н.Н Биккулова, Г.Н. Асылгужина, А.Н Скоморохов и др. ” Кристаллическая структура и ионный перенос в  $\text{AgCuSe}$ ” // Изв. РАН. Сер. Физ. 2006.Т. 70. С.559.
- Л6. Н.В.Астахова, И.М.Саламатин, В.Н.Швецов. “Программный комплекс АС (Автоматизация спектрометрии). 1. Концепция программной системы, инвариантной по отношению к изменениям методики эксперимента” // ПТЭ, 2004, №5, с.56-61.