

P13-2003-145

Н. В. Астахова, И. М. Саламатин, В. Н. Швецов

**ПРОГРАММНЫЙ КОМПЛЕКС АС
(АВТОМАТИЗАЦИЯ СПЕКТРОМЕТРИИ)
Концепция программной системы,
инвариантной по отношению к изменениям
методики эксперимента**

Направлено в журнал «Приборы и техника эксперимента»

максимальном - обеспечивает также и полную математическую обработку данных одновременно с регистрацией. Такая САЭ должна обеспечивать:

- унифицированный графический интерфейс пользователя, надежность, простоту управления, возможность экспресс-анализа регистрируемых данных;
- средства локального и дистанционного управления, визуального контроля состояния спектрометра и экспериментальных данных;
- возможность подключения on-line визуализации и обработки экспериментаторами путем простой настройки системы;
- унифицированный способ сохранения данных, поступающих от подсистемы регистрации; в дальнейшем предполагается совместимость с форматами, принятыми в других исследовательских центрах;
- наращивание функциональных возможностей одновременно для ряда спектрометров и обеспечение преемственности ранее разработанных методик математической обработки данных;
- автоматическое составление описаний условий регистрации каждого файла данных;
- протоколирование для каждого файла информации о событиях, влияющих на качество экспериментальных данных; автоматический анализ качества данных (совместимости с остальными данными);
- возможность автоматического восстановления работоспособности системы с минимальной потерей данных при сбоях, временной потере электропитания и др.;
- инвариантность программного обеспечения относительно изменений методики физического эксперимента;
- максимально возможную гибкость программного обеспечения относительно изменений конфигурации спектрометра.

Для разработки таких систем автоматизации эксперимента в ограниченные сроки требуется поиск специальных решений. Мы полагаем, что некоторые трудности разработки ПО, а также ожидаемые проблемы эксплуатационного характера отчасти вызваны способом учета методики эксперимента. Сокращения объема работ и сроков разработки можно ожидать, если мы ограничим область применения ПО условиями (см. выше), характерными для нашей организации, формализуем описание методики

эксперимента, разделим ПО на две части: зависящую от методики (драйверный уровень) и инвариантную к ее изменениям (базовое ПО).

В данной статье описывается вариант применения такой формализации в области задач спектроскопии по времени пролета с буферизацией данных.

1. Программное обеспечение спектрометра

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

- набора источников экспериментальных данных (детекторов);
- набора блоков памяти, в которые данные от источников (детекторов) заносятся тем или иным способом («первичная» память);
- набора устройств, изменяющих условия эксперимента (терморегуляторы; двигатели перемещения мишеней, детекторов; спин-флипперы и др.).

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

- инициализацию (настройку) детекторной и регистрирующей электроники в соответствии с разработанной структурой данных; при этом требуется:
 - задать «аппаратные» адреса инициализируемого оборудования и
 - способ группирования операций управления потоком данных;
 - задать состав и структуру записываемых в первичную память данных (длина слова, размерность, последовательность осей) и др.
- управление потоком данных в первичную память с операциями «Чтение», «Запись», «Старт», «Стоп» и др.;
- управление устройствами, изменяющими условия получения экспериментальных данных.

Эта группа программ должна устанавливать необходимые значения из области допустимых для конкретного устройства. В дальнейшем будем называть такой состав программ драйверным уровнем программного обеспечения спектрометра. Необходимы также программы настройки спектрометра (выбор оптимального положения образца, коллиматоров и т.д.), тестовые и демонстрационные блоки программ, построенные с использованием указанных выше трех групп программ.

В любом проекте ПО драйверного уровня создается в том или ином виде, и от его свойств, качества, состава и структуры в значительной мере зависит судьба проекта системы автоматизации эксперимента.

Отметим следующее: структура регистрируемых данных зависит от методики эксперимента. Поэтому целесообразно на начальном этапе разработки ПО детекторной и регистрирующей электроники ввести способ учета возможных вариантов структур данных, например, посредством декларативных инструкций. В этом случае базовое ПО спектрометра может быть реализовано в виде, инвариантном относительно изменений методики эксперимента. Такие декларации необходимы также и для подсистемы обработки данных (см. ниже). Мы полагаем, что в данной части наши взгляды близки к изложенным в работе [5]. Различается способ реализации программного обеспечения драйверного уровня: в [5] используется система LabView, а мы в этой части не ограничиваемся одним стандартом.

Описываемый в данной работе подход к построению подсистемы регистрации и САЭ допускает использование драйверного ПО спектрометра, зависящего от методики эксперимента. Следствием такой зависимости будет некоторое снижение эффективности предлагаемого подхода, дополнительное программирование и ограничения, существенность которых зависит от конкретной реализации используемого ПО.

2. Программное обеспечение эксперимента

В процедуре проведения эксперимента обычно выделяются следующие задачи:

- регистрация и накопление экспериментальных данных (обеспечение требуемой статистики);
- предварительная обработка (сортировка, экспресс-анализ, отбраковка, свертка
- данных, визуализация, вывод информации на носители, математические операции с массивами по заданной в методике формуле и т.п.);
- полная математическая обработка и анализ данных.

Первая задача является наиболее важной и зачастую отождествляется с задачей автоматизации эксперимента. Рассмотрим способы реализации методики эксперимента на этапе регистрации экспериментальных данных.

2.1. Подсистема регистрации данных

Подсистема регистрации может быть выполнена в виде универсальных базовых программ, являющихся надстройкой над драйверным ПО спектрометра. Для рассмотрения варианта такой надстройки введем понятие параметрической модели спектрометра. Под такой моделью будем понимать список параметров, характерных для данного спектрометра и определяющих возможные условия проведения эксперимента. В терминологии физика - это, например, температура мишени, положение детектора, состояние спин-флиппера, номер мишени, а также некоторые параметры, необходимые для обработки данных.

Основа обсуждаемой надстройки - четыре программы, каждая из которых не зависит от методики конкретного эксперимента:

1. Программа диалога с физиком, в котором он может определить список параметров для данного эксперимента, задать значения этих параметров (для каждого параметра - свой вектор значений) и очередность их изменения. Другими словами - задать матрицу значений изменяемых параметров. Подготовка таких данных физиком предполагается в любом варианте построения программы эксперимента, в данной работе лишь предлагается представить их, например, в виде разреженной матрицы или списковой структуры. С позиций информатики такой набор данных является формальным описанием методики конкретного эксперимента.
2. Программа, выбирающая из матрицы значения вектора параметров (температура, состояние спин-флиппера, номер мишени, ...) для очередного измерения. Такой вектор, совместно со значениями неизменных параметров, дает в некоторой обобщенной системе координат полное описание состояния системы, выполняющей измерение. Будем для краткости называть его вектором состояния. В этом векторе значение каждого параметра определено с учетом заданной физиком очередности изменения параметров.
3. Программа, устанавливающая условия измерения, используя для этих целей вектор значений параметров очередного измерения и готовые программы драйверного уровня ПО спектрометра, список которых может быть включен в описание параметрической модели.
4. Программа, реализующая обратную связь между подсистемами регистрации и обработки данных. Обратная связь может потребоваться в случае обнаружения в

процессе эксперимента изменения эффективности детектора, положения реперных пиков и т.п. Помимо этого, она выполняет регистрацию факта завершения работы с текущим вектором состояния, и это обеспечивает возможность возобновления эксперимента при аварийных сбоях операционной системы, системы питания, сброса мощности и др. при минимальных потерях накопленной к этому моменту информации.

В общем случае эти четыре программы обеспечат привязку программного обеспечения к методике конкретного эксперимента, а также к конфигурации спектрометра. Разумеется, не исключается и возможность разработки более специфических программ, если к тому будут мотивы.

Заметим, что разработка существенно облегчается тем фактом, что ни одна из них не работает непосредственно с экспериментальным оборудованием.

Построение такой привязки к методике эксперимента и составу оборудования посредством группы унифицированных базовых программ позволяет на всех спектрометрах использовать одну и ту же программу управления потоком данных (подсистему регистрации), работающую по следующему алгоритму:

- выбрать из матрицы параметров очередную строку - вектор состояния;
- установить в спектрометре условия в соответствии с выбранным вектором состояния;
- выполнить экспозицию - измерить спектры;
- отправить измеренные данные в архив или на обработку, зарегистрировать факт завершения работы с текущим вектором состояния;
- повторять предыдущие шаги, пока не исчерпается матрица параметров.

На рис. 1 приведена упрощенная блок-схема системы автоматизации эксперимента, в которой реализованы изложенные выше принципы. Эта система построена на основе программ в формате *.exe*, взаимодействующих друг с другом посредством обмена командами, данными и сообщениями. Такой способ построения систем используется в ряде работ и имеет хорошие технологические свойства. Он допускает выполнение отдельных частей комплекса разными разработчиками независимо, дает возможность выполнять разработку на любых языках

программирования и автономную проверку. Возможно также использование отдельных компонентов в отрыве от всего комплекса.

Следует отметить, что такая программа управления экспериментом может работать в безостановочном режиме: после исчерпания матрицы параметров управляющая программа может получить следующее задание (матрицу) по сети или другим способом.

2.2. Сравнение с “традиционным” подходом к построению ПО эксперимента

На сегодняшний день в ряде организаций [4,6,7] используется путь построения ПО эксперимента, который можно считать традиционным:

- Создание драйверного ПО спектрометра.
- Разработка (либо использование готового) языка описания предполагаемых операций экспериментатора с оборудованием спектрометра и данными.
- Разработка программы-интерпретатора, понимающей этот язык.

На этом языке пишется программа эксперимента, вернее - шаблон программы. В такой программе, в конечном счете, должны появиться конкретные значения всех изменяемых параметров. Шаблон программы модифицируется для каждого последующего эксперимента со всеми вытекающими из такого сценария работы последствиями. Очевидно, что при изменениях в методике эксперимента должен быть создан новый шаблон и выполнена его отладка, т.е. преемственность отсутствует. По мере уточнения представлений о том, что на самом деле требуется экспериментатору от данного спектрометра, язык и интерпретатор эпизодически модифицируются, что может привести если не к потере прежних программ-шаблонов, то, по крайней мере, к необходимости их проверки.

В отличие от "традиционного", предлагаемый подход вместо программирования процедуры эксперимента использует метод, который можем пока назвать "методом описания состояний установки". Данный метод использует базовое ПО спектрометра и программы привязки к методике, не зависящие от самой методики эксперимента, а также в минимальном объеме аппаратно-зависимую часть - программы драйверного уровня спектрометра.

Следует подчеркнуть, что суть предложения не в отказе от использования интерпретаторов, а в специальной структуре программы управления регистрацией. Мы

предлагаем операции привязки к методике эксперимента переместить на более высокий уровень. В результате предложенная структура, в которой используется автоматически обрабатываемый вектор состояния спектрометра, дает нам ряд технологических и функциональных возможностей, трудно реализуемых либо недоступных в традиционном подходе. Одна и та же группа управляющих и функциональных программ может использоваться на разных спектрометрах для разных методик. Благодаря такой прозрачности программ мы получаем простоту освоения и использования, сокращение затрат на их разработку, преемственность, возможность реализации измененной методики эксперимента физиками без участия программистов до тех пор, пока их устраивает состав имеющегося оборудования спектрометра. Включение в методику эксперимента параметра нового типа (конечно, при наличии соответствующего оборудования) решается созданием двух программ. Эти программы - драйвер дополнительного устройства и (в редких случаях) программа описания возможных значений (точнее - обобщенных значений) управляемого параметра для диалога с экспериментатором при составлении матрицы значений изменяемых параметров.

2.3. Подсистема обработки

Известны системы, способные выполнять в режиме работы на линии с установкой регистрацию и предварительную обработку [8,9] либо полную обработку [10] на управляющей ЭВМ. Однако в большинстве случаев по различным причинам задачи регистрации, предварительной и окончательной обработки данных реализуются с разрывом во времени, и программное обеспечение создается разными группами специалистов [4].

Границу между "предварительной" и "окончательной" обработкой данных установить затруднительно. Можно полагать, что к предварительной обработке относятся операции, выполняемые в режиме OnLine, к окончательной - операции, выполняемые с использованием специальных программ после завершения эксперимента. При этом место и способ выполнения операций согласования форматов представления результатов предварительной обработки с требованиями специальных программ определяются конкретной реализацией.

Доступные в настоящее время ресурсы оперативной памяти, накопителей на жестких дисках и производительность процессоров позволяют увеличить число

операций обработки, выполняемых в автоматическом режиме одновременно с регистрацией данных.

В данной работе предлагается способ построения единой подсистемы обработки данных (подсистема ОД) в составе САЭ. Процесс обработки разделяется на 3 фазы:

- 1) фильтрация, сжатие и архивация данных;
- 2) предварительная обработка - получение нормализованных данных;
- 3) использование программ полной математической обработки данных.

В данной работе введено понятие вектора состояния, формируемого и обрабатываемого автоматически. Использование в системе такой структуры данных позволяет состыковать в автоматическом режиме работу подсистем регистрации и обработки данных. Использование ОД в автоматическом режиме позволяет решить проблемы преемственности готовых программ обработки данных.

На рис. 2 представлена схема подсистемы обработки данных. Пунктирными линиями разделены группы операций, выполняемые на трех различных этапах работы подсистемы ОД.

2.3.1. Фильтрация, сжатие и архивация данных. В спектрометрии типичной практикой, используемой для накопления статистически достоверных данных, является повторение измерений одного типа (имеющих одинаковый вектор состояния). Поэтому, в общем случае, мы получаем поток однотипных данных для каждого вектора состояния, которые предстоит просуммировать. Из подсистемы регистрации в ОД поступает блок данных от всех источников, управляемых совместно. В числе этих данных спектральная информация, счет мониторов мощности источника излучения, счет таймера продолжительности экспозиции, а также аварийная или предупреждающая информация для организации обратной связи и протоколирования.

Необходимая операция на данном этапе работы ОД - занесение в вектор состояния характеристик потока излучения (счет мониторов, продолжительность экспозиции и др.). Используя вектор состояния, мы можем вычислить названия файлов данных и вектора состояния (либо другой дескриптор) для записи их в архив. Вектор состояния неотделим от блока спектральной информации.

На рис. 2 приведена схема обработки потока данных определенного типа (т.е. данных для одного вектора состояния). Из рисунка видно, что ОД использует две программы:

1. Программа "Фильтр" для обработки каждого блока данных вызывает по очереди функции, названия которых занесены в список. Эти функции выполняют сортировку данных, занесение в архив, сглаживание, преобразование осей, поканальное суммирование, вывод данных в подсистему OnLine-визуализации и др. операции в соответствии с заданной методикой обработки данных. Функции фильтрации, используемые на данном этапе обработки, могут и должны быть исполнены в виде, не зависящем от типа спектрометра;
2. Программа "Контроль" тоже использует список функций. Эти функции выполняют контроль постоянства эффективности детектора, проверяют неизменность фоновых условий, положение реперных пиков, обрабатывают калибровочные измерения и т.п. Результаты работы функций контроля используются для отделения несовместимых данных при архивации и в качестве сигнала обратной связи с подсистемой регистрации. В числе операций контроля могут оказаться такие, которые требуются только в одном конкретном эксперименте.

Результаты работы подсистемы ОД на этапе сжатия данных - это вычисленные спектры и соответствующие им векторы состояния. Программы при выполнении арифметических операций с данными должны вводить необходимую коррекцию и в вектор состояния. Так, например, при суммировании данных должны суммироваться счет мониторов и времена экспозиции.

2.3.2. Получение нормализованных спектров. Обычно для полной математической обработки используются специально разрабатываемые программы, алгоритмы которых отражают способ получения информации об исследуемом физическом явлении. Для нас существенно, что эти программы в качестве исходных данных используют определенный набор некоторым образом нормализованных спектров разного назначения и значения ряда параметров, которые в нашем случае присутствуют в соответствующем векторе состояния. Способ получения этих нормализованных данных и их перечень определяются методикой подготовки данных для конкретной программы полной обработки данных.

Физический смысл "нормализации" спектров заключается в преобразовании шкал осей. В некоторых случаях, например для оси времени (в случае времяпролетной методики), переход к энергетической шкале или шкале длин волн сводится к пересчету координат точек по формулам, все параметры которых известны до начала измерения.

Иначе обстоит дело с осью счета в каналах, отражающей измеряемый эффект, искаженный влиянием условий измерения. Для преобразования аппаратного спектра (счет в каналах) в спектр сечения, пропускания, отражения и т.д. требуется использование дополнительных экспериментальных данных. Поэтому, одновременно с измерениями, непосредственно предназначенными для исследования физического явления, выполняются вспомогательные измерения спектра фона, формы потока излучения, калибровка и др. Вспомогательные измерения могут выполняться и в отдельных экспериментах. Назначение каждого измерения (спектр фона, потока, ...) известно экспериментатору в момент составления матрицы параметров. Такую информацию следует включать в список параметров, если предполагается автоматическая совместная обработка спектров разного назначения или автоматизированный поиск соответствующих данных.

На рис.2 показан для примера вариант нормализации спектров для спектрометра SPN [11]. В этой схеме предполагается, что на вход программы поступает список спектров, измеренных при различных векторах состояния. Данная программа выбирает из списка спектр $S[\text{vec}(S,f)]$, соответствующий вектору состояния $\text{vec}(S,f)$ с состоянием f спин-флиппера, и выполняет их обработку совместно с одними и теми же спектрами фона F и потока нейтронов B , измеренными ранее при отсутствии мишени. Продолжительность экспозиции может быть извлечена из вектора функцией $\text{tim}(\text{vec}(S,f))$. В процессе обработки требуется привести все спектры к одинаковому времени измерения $\text{tim}(\text{vec}(S,f))$, вычесть фон из спектров S и B , выполнить поканальное деление одного спектра на другой и для полученного массива отношений вычислить массив статистических ошибок E . В данном примере требуются два выражения - формула вычисления коэффициента отражения $R[\text{vec}(S,f)]$ (1) и погрешностей (вторая не приводится для краткости изложения):

$R[\text{vec}(S,f)] =$

$$\{S[\text{vec}(S,f)]/\text{tim}(\text{vec}(S,f)) - F[\text{vec}(F,0)]/\text{tim}(\text{vec}(F,0))\} / \{B[\text{vec}(B,0)]/\text{tim}(\text{vec}(B,0))\}; \quad (1)$$

Такую обработку можно выполнять с помощью, например, интерпретатора матричной арифметики, принимающего аналитическое выражение в качестве описания алгоритма работы [12]. В последнее время в системах моделирования процессов [13] развивается более гибкий интерфейс, чем язык интерпретатора - использование

табличных или списковых структур, управляемых потоком данных и программными событиями. Однако в данном случае важным является не способ вычисления по формуле, которая меняется достаточно редко, а возможность удобно менять состав используемых данных.

Следует отметить, что включение в состав ОД программы, реализующей подготовку нормализованных спектров, сокращает продолжительность цикла обработки данных.

2.3.3. Полная математическая обработка и анализ результатов. Выполнение полной математической обработки в режиме OnLine целесообразно для получения оценочных результатов и организации обратной связи с подсистемой регистрации данных в автоматическом либо диалоговом режиме. В основном же это процесс итерационный, требующий активного участия экспериментатора.

Подсистема ОД с предложенной структурой и свойствами позволяет выполнять обработку в Turbo-режиме. Эффективность Turbo-режима известна на примерах современных средств разработки программ, для которых его наличие стало стандартом. В данном предложении реализация Turbo-режима дает возможность после выполнения обработки и анализа результатов:

- изменить состав функций фильтрации, контроля данных (т.е. изменить состав используемых данных из числа измеренных и зарегистрированных в архиве);
- указать другие вспомогательные спектры (файлы фона, формы потока и др.);
- повторить в автоматическом режиме процесс обработки данных.

Схема на рис. 2 поясняет способ реализации и использования Turbo-режима в данной подсистеме. Из рисунка видно, что реализация Turbo-режима, как и всей подсистемы ОД (за исключением уникальной программы полной обработки), обеспечивается довольно простыми программами, большая часть которых инвариантна к методике эксперимента и типу спектрометра.

Программы, включенные в состав подсистемы ОД и используемые несколькими группами экспериментаторов, могут располагаться и исполняться на сервере.

В предложенной структуре подсистемы ОД обеспечена гибкость в отношении изменения алгоритмов обработки и организации диалогового режима работы экспериментатора. При устоявшейся методике обработки алгоритмы могут не меняться более 10 лет (например, [10]), поэтому заложенная избыточная гибкость имеет целью получить технологический эффект на этапе доработки методики и при разработке

программ для различных спектрометров. Что же касается способа организации диалогового режима, то можно ожидать более интенсивное использование turbo-режима.

3.Выбор операционной системы

ОС Windows (9X, NT, 2000 и др.) не является операционной системой реального времени. В результате этого время переключения процессов, а также скорость и способ передачи данных не соответствуют требованиям ряда задач реального времени [14]. Известны работы, результатом которых является устранение этих затруднений, однако при этом мы получаем фактически другую операционную систему [14].

Штатные версии ОС Windows 9X, NT, 2000 и др. предоставляют огромные возможности для быстрой разработки приложений, которыми, естественно, хочется воспользоваться. Этот интерес укрепляется в связи с быстрой реакцией фирмы Microsoft на возникновение новых идей и продуктов и распространенностью данной ОС.

Особенностью САЭ в области спектрометрии, использующих буферизацию данных в первичной памяти (например, [1,2]), является отсутствие жестких требований ко времени переключения процессов и умеренная загрузка процессора. Средствами штатных версий ОС Windows с использованием дополнительных разработок на прикладном уровне можно получить:

- среднее время выполнения запроса к ОС для синхронизации процессов (Get mutex) составляет 35 мкс, максимальное - несколько мс [14];
- время реакции программы пользователя на внешние запросы прерывания ~150 мкс *;
- время передачи команд или сообщений от задачи к задаче 150 мкс *;
- время доступа к разделяемой (общей) памяти 50 мкс *;
- время записи в разделяемую память с передачей сообщения об этом другому процессу (генерация события) 200 мкс *.

Такие характеристики достаточны для реализации описанного ПО САЭ. Предлагаемая система может рассматриваться как непрерывно работающий компонент

* Pentium II, 450 мгтц.

системы более общего назначения, что дает нам дополнительные основания интересоваться возможностью работать в штатной среде ОС Windows.

Интерес фирмы Microsoft к ОС Linux и появление под ОС Windows соответствующих кросс-средств разработки облегчает задачу реализации описанной САЭ для обеих ОС.

4. Основные результаты

1. Для спектрометров, буферизирующих данные в памяти, разработана специальная структура и алгоритм работы подсистемы регистрации данных, использующей базовое программное обеспечение и программы драйверного уровня. Введены понятия параметрической модели спектрометра, вектора состояния системы регистрации и операции над векторами состояния. Предложено в каждом измерении учитывать методику эксперимента путем задания вектора состояния системы регистрации в терминах изменяемых параметров вместо программирования операций управления. Базовое программное обеспечение, построенное в соответствии с изложенными принципами, пригодно для использования без изменений в различных экспериментах на различных спектрометрах и не налагает ограничений на реализацию методики эксперимента.

2. Разработана структура программного обеспечения системы автоматизации эксперимента, в рамках которой можно реализовать:

- предварительную и полную математическую обработку экспериментальных данных в режиме работы на линии с оборудованием;
- turbo-режим обработки, автоматизирующий циклическое выполнение привычного процесса:
 - изменение состава обрабатываемых данных и операций их преобразования,
 - выполнение вычислений,
 - анализ полученных результатов.

Использование предложенного способа описания методики для группы различных экспериментов дает сокращение объема необходимого программирования при одновременном увеличении реализуемых возможностей.

Фрагменты описанной системы использовались в течение ряда лет в ПО ряда спектрометров [8,9,10,15]. В данной работе учитывается полученный опыт.

Работа выполнена в ЛНФ имени И.М. Франка и ориентирована в первую очередь на спектрометрические эксперименты, выполняемые на ИБР-2, однако некоторые из предлагаемых решений могут найти и более широкое применение.

Список литературы

1. *Korobchenko M.L. et al.* Unified VME-Based Data Acquisition Systems for the Spectrometers at the IBR-2 Pulsed Reactor. Processor Module // Proceedings of the International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-97), Dubna, 2-4 June 1997, JINR, E10-97-272, Dubna, 1997, p. 163-171.
2. *Korobchenko M.L. et al.* The VME Spectrometric Memory Unit // Proceedings of the International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-97), Dubna, 2-4 June 1997, JINR, E10-97-272, Dubna, 1997, p. 213-217.
3. *Furman W.I.* New pulsed neutron source of JINR-the IREN project.// Proceedings of the 3-rd International Conference, Casta-Papiernicka, Slovak Republic, Aug. 30-Sept 4, 1996. In 'Dynamical Aspects of Nuclear Fission', edited by J. Kliman, B.I. Pustilnik, Dubna 1996, p. 394.
4. *Filhol A. et al.* Data Visualisation and Data Processing at the ILL. Proceedings of the International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-97), Dubna, 2-4 June 1997, JINR, E10-97-272, Dubna, 1997, p. 231-243.
5. *Wright M.C., Hubbard C.R., Lenarduzzi R. And Rome J.A.* Internet-Based Remote Collaboration at the Neutron Residual Stress Facility at HFIR.// Proceedings of the NOBUGS-2002, Gaithersburg, MD, USA, Nov.4-6, 2002.
6. *Astachov Yu. et al.* Current State and Prospects for Developing the FLNP Measurement and Computation Complex // Proceedings of the International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-97), Dubna, 2-4 June 1997, JINR, E10-97-272, Dubna, 1997, p. 153-162.
7. *Kirilov A.* Current State and Perspectives of the IBR-2 Instrument Control Software // Proceedings of the Second International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-2000), Dubna, 5-7 June 2000, JINR, E10-2001-11, Dubna, 2001, p. 206-210.
8. *Балука Г. и др.* Препринт ОИЯИ Р10-12960, Дубна, 1980.

9. *Вагов В.А. и др.* Измерительный модуль спектрометра малоуглового рассеяния нейтронов на импульсном реакторе ИБР. Препринт ОИЯИ Р10-80-826, Дубна, 1980.
10. *Ostanevich Yu.M.*, Macromol. CHEM; Macromol Symp. 15, 91-103 (1988). Time-of-Flight Small-Angle Scattering Spectrometers on Pulsed Neutron Sources.
11. *Журавлев В.В. и др.* Измерительно-накопительный модуль спектрометра на поляризованных нейтронах СПН-1. Препринт ОИЯИ, Р3-95-140, Дубна, 1995.
12. <http://www.mathworks.com/products/matlab>
13. <http://www.mathworks.com/products/simulink>
14. *Timmerman M., Monfret J-C.*, Windows NT in Real-Time System, the Industry Dream? Report at X-th IEEE Real Time Conference, Beaune, France, 22-26 September, 1997.
15. *Astakhova N.V. Kirilov A.S, Salamatin I.M.* Remote Control of the YUMO Spectrometer and User Interface // Proceedings of the Second International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-2000), Dubna, 5-7 June 2000, JINR, E10-2001-11, Dubna, 2001, p. 275-278.

Получено 24 июля 2003 г.

Астахова Н. В., Саламатин И. М., Швецов В. Н.
Программный комплекс АС (автоматизация спектрометрии).
Концепция программной системы, инвариантной по отношению
к изменениям методики эксперимента

P13-2003-145

Для спектрометрических экспериментов, буферизирующих данные в памяти, разработана специальная структура и алгоритм работы подсистемы регистрации данных, использующей универсальное базовое программное обеспечение и программы драйверного уровня для конкретного спектрометра. Предложено в каждом измерении учитывать методику эксперимента путем задания вектора состояния спектрометра в терминах изменяемых параметров вместо традиционного программирования операций управления спектрометром. Такая программа подсистемы регистрации пригодна для использования без изменений на различных спектрометрах и не налагает ограничений на реализацию методики эксперимента.

Разработана структура программного обеспечения системы автоматизации эксперимента, в которой можно реализовать предварительную (и полную) математическую обработку экспериментальных данных в режиме работы на линии с оборудованием и turbo-режим обработки, автоматизирующий циклическое выполнение операций редактирования состава используемых данных, вычисления, анализа результатов.

Использование предложенного способа учета методики для групп различных экспериментов дает сокращение объема необходимого программирования при одновременном увеличении реализуемых возможностей.

Работа выполнена в Лаборатории нейтронной физики им. И. М. Франка ОИЯИ.
Препринт Объединенного института ядерных исследований. Дубна, 2003

Перевод авторов

Astakhova N. V., Salamatin I. M., Shvetsov V. N.
Software Complex AS (Automation of Spectrometry).
Conception of a Program System Invariant to Experiment Method Variation

P13-2003-145

For spectrometric experiments in which data are buffered in the memory, a special structure and operation algorithm of the data registration subsystem employing the basic software of the spectrometer has been developed. It suggests accounting for the experimental method in every measurement by specifying the vector of the registration system state in terms of varied parameters instead of traditional programming of spectrometer control. This type of program for registration subsystem can be used for different spectrometers without modification and it does not impose restrictions on the method of realization of the experiment.

A software structure for the experiment automatization system has been developed, which allows realization of complete mathematical processing of the experimental data on-line as well as of turbo-mode processing that automates cyclic performance of such operations as editing of the experimental data content used, computation, and results analysis.

The use of the proposed method of accounting for the experimental method to carry out a group of different experiments will reduce the volume of programming, enlarging the realized possibilities.

The investigation has been performed at the Frank Laboratory of Neutron Physics, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna, 2003

Редактор *М. И. Зарубина*
Макет *Н. А. Киселевой*

Подписано в печать 21.08.2003.

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

Усл. печ. л. 1,0. Уч.-изд. л. 1,3. Тираж 315 экз. Заказ № 54053.

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

E-mail: publish@pds.jinr.ru

www.jinr.ru/publish/