

P13-2007-93

Н. Г. Мазный¹, И. М. Саламатин, К. М. Саламатин²

ГЕНЕРАЦИЯ ПРОГРАММ АВТОМАТИЗАЦИИ
ЭКСПЕРИМЕНТОВ ИЗ МОДУЛЕЙ В ФОРМАТЕ
ЗАГРУЗКИ

Направлено в оргкомитет 57-й международной конференции
«Ядро-2007», Воронеж, 25–29 июня 2007 г.

¹НПЦ «Аспект»; E-mail: nikitos@dubna.aspect.ru

²ГУ «Дубна»

Мазный Н. Г., Саламатин И. М., Саламатин К. М.

P13-2007-93

Генерация программ автоматизации экспериментов из модулей
в формате загрузки

В работе поставлена задача существенного сокращения сроков разработки и модификации САУ. Разработанная структура САУ включает подсистему подготовки задания, программу управления работой (интерфейс пользователя), подсистему управления состоянием управляемого объекта и подсистему обработки данных. Процессы подготовки задания и его исполнения разделены. При наличии готовых программных модулей интегрирование и модификацию САУ может выполнить пользователь.

Работа выполнена в Лаборатории нейтронной физики им. И. М. Франка ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна, 2007

Maznyj N. G., Salamatin I. M., Salamatin K. M.

P13-2007-93

Generation of Experiment Automation Programs from Modules
in Loadable Format

The work aims at searching for methods to significantly decrease periods of ACS development and modification. The developed ACS structure includes task preparation subsystem, operation control program (user's interface), subsystem to control the state of monitored object (MO), and data processing subsystem. Task preparation and its execution are arranged as separate processes. Provided that ready program modules are available, ACS integration and modification can be performed by the user.

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

Preprint of the Joint Institute for Nuclear Research. Dubna, 2007

ВВЕДЕНИЕ

Системы автоматизированного управления (САУ) в исследовательских организациях в течение срока жизни все чаще подвергаются изменению относительно методики эксперимента, состава управляемых параметров (состава оборудования). Расход времени на модификацию САУ приводит к задержке решения нужных задач и простоя ценного оборудования. Поэтому актуальность методов сокращения сроков разработки, тем более — адаптации САУ к новым условиям использования, растет. Одна из очевидных причин удлинения сроков разработки САУ — в необходимости использовать элементы программирования на этапе адаптации САУ к новой методике и конфигурации оборудования.

В данной работе поставлена задача поиска методов существенного сокращения сроков разработки и модификации САУ. Предлагаемый способ решения этой задачи — компоновка САУ из унифицированных базовых модулей в формате загрузки, автоматическая настройка этих модулей и интерфейса пользователя на конфигурацию САУ по ее текстовой документации и предоставление пользователю возможности модифицировать САУ, не используя средства программирования.

1. СТРУКТУРА САУ И АЛГОРИТМ УПРАВЛЕНИЯ

Абстрагируемся от деталей процесса управления. 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. 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.

Для дальнейшего рассмотрения введем понятие параметрической модели объекта как списка параметров, характерных для текущей конфигурации и

определяющих набор возможных прикладных методик. Это могут быть, например, температура, номер мишени, положение детектора, порог аварийной сигнализации гамма-датчика, продолжительность экспозиции данных и т. п.

Отделим процесс подготовки задания САУ от его выполнения. Введем диалоговую программу, с помощью в которой пользователь 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 простой структуры, содержащей только названия параметров и нужные значения. С позиций информатики такой набор данных является формальным описанием прикладной методики для конкретного сеанса работы, строка матрицы описывает условия очередного измерения.

Структуру ПО САУ представим в виде нескольких универсальных базовых программ. Основа обсуждаемой структуры — это три программы, каждая из которых не зависит от конкретной прикладной методики:

1. Программа, выбирающая из матрицы задания строку и передающая ее исполняющей подсистеме для выполнения очередного измерения. Такая строка является вектором состояния САУ. По сути, данная программа является **программой управления работой САУ**.
2. Программа, устанавливающая условия измерения, используя для этих целей выбранный вектор параметров очередного измерения и готовые программы драйверного уровня ПО установки. Данная программа — **программа управления состоянием** — является основным модулем в исполняющей подсистеме САУ.
3. **Подсистема предварительной обработки** данных выполняет ряд системных функций общего назначения, а также может вызывать функции обработки данных, состав и последовательность выполнения которых определяются 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 и восстанавливаемых) потерях накопленной к этому моменту информации и др.

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

- выбрать из матрицы параметров очередную строку — вектор состояния;
- установить на объекте условия в соответствии с вектором состояния и выполнить измерения;
- отправить данные в архив или на обработку, зарегистрировать факт завершения работы с текущим вектором состояния;
- повторять предыдущие шаги, пока не исчерпается матрица параметров.

Структура САУ, соответствующая этому алгоритму, включает подсистему подготовки задания, программу управления работой (интерфейс пользователя), подсистему управления состоянием управляемого объекта (УО) и подсистему обработки данных (см. рис. 1).

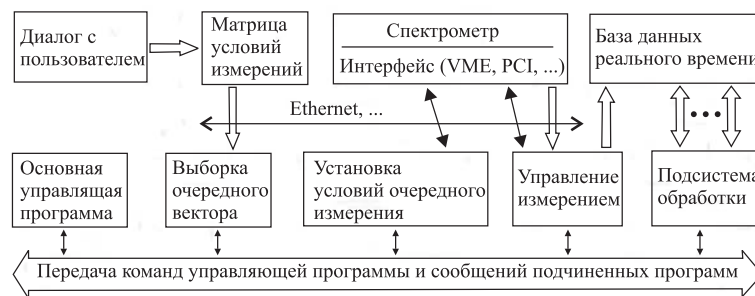


Рис. 1. Упрощенная блок-схема системы автоматизированного управления

Прозрачность алгоритма дает возможность пользователям варьировать прикладную методику без участия программистов до тех пор, пока их устраивает состав имеющегося оборудования (состав параметров) объекта автоматизации.

Включение в прикладную методику параметра нового типа, не предусмотренного ранее в параметрической модели, означает расширение конфигурации САУ. Чтобы обеспечить возможность расширения конфигурации без привлечения средств программирования, потребовалось

- разработать подсистему подготовки задания, автоматически настраивающуюся на нужную конфигурацию САУ (и методику работы);

- разработать специальную структуру компонентов драйверного слоя, обеспечивающую автоматическое подключение их к подсистеме подготовки задания и исполняющей системе.

2. ПОДСИСТЕМА ПОДГОТОВКИ ЗАДАНИЯ

Традиционный способ управления работой САУ — использование интерфейса пользователя, специально разрабатываемого для конкретной САУ [1, 2], позволяющего задать значения параметров исполняющей системе. В этом случае набор доступных пользователю методик определяется фиксированным кодом интерфейса, и выход за эти рамки требует изменения программ. Альтернативный вариант — это единый для различных САУ интерфейс пользователя, автоматически настраивающийся на нужную методику. Рассмотрим вариант такого интерфейса. В состав подсистемы подготовки задания введем базу данных САУ и программу подготовки задания **PSJ** (Preparation of Single Job).

2.1. База данных САУ. В базе данных введены таблицы:

- **описания компонентов** драйверного слоя, доступных разработчикам в данной организации, — «библиотека» устройств;
- **описание конфигурации** драйверного слоя разрабатываемой САУ;
- **таблица изменяемых параметров;**
- таблица **дополнительных параметров**, используемых в программах обработки данных, при подготовке описаний данных, отчетов и др.;
- список **наблюдаемых параметров**, значения которых требуется визуализировать с заданной периодичностью;
- три **таблицы заданий**: новых, отложенных, завершенных.

Таблицы изменяемых и дополнительных параметров составляют параметрическую модель управляемого объекта.

Описание компонента драйверного слоя содержит:

- текстовую документацию, необходимую пользователю для решения вопроса о возможности его использования для решения нужной задачи;
- список заданных программистом названий параметров, их типы, диапазоны значений и значения, принимаемые по умолчанию.

Помимо программ управления реальными устройствами, в число компонентов драйверного слоя мы включаем также функциональные программы, исполнение которых влияет на состояние САУ.

На рис. 2 показано окно интерфейса пользователя, предназначенное для заполнения библиотеки устройств, — списка доступных разработчику компонентов.

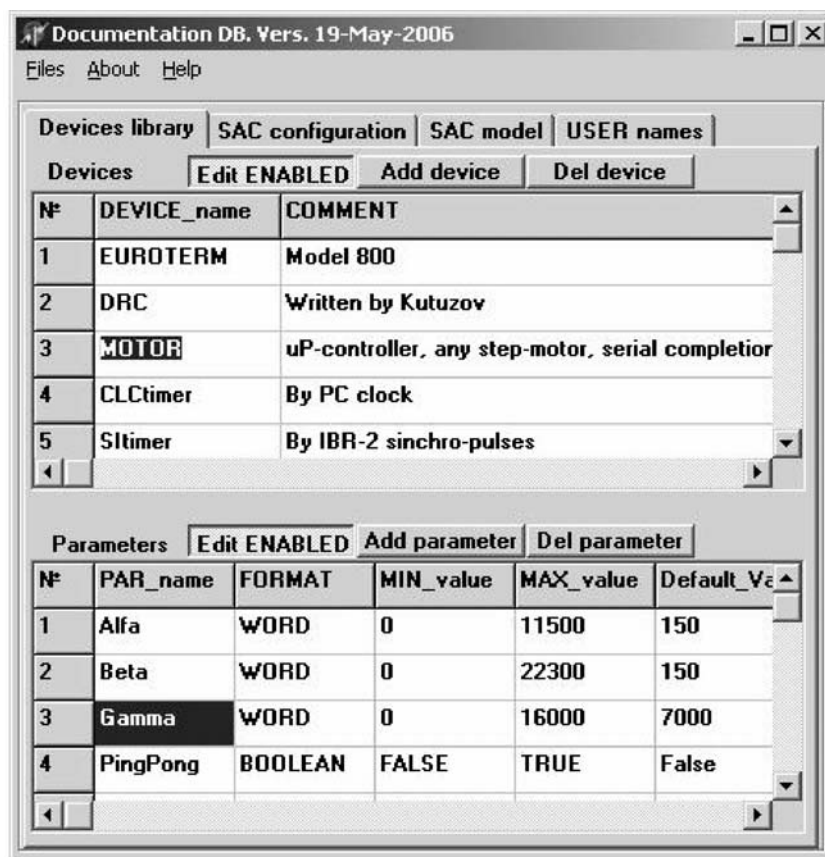


Рис. 2. Окно интерфейса пользователя, предназначенное для заполнения библиотеки устройств в базе данных

Описание состава драйверного слоя (см. рис. 3) включает перечень используемых в данной САУ программных компонентов и информацию о способе их подключения (название контроллера).

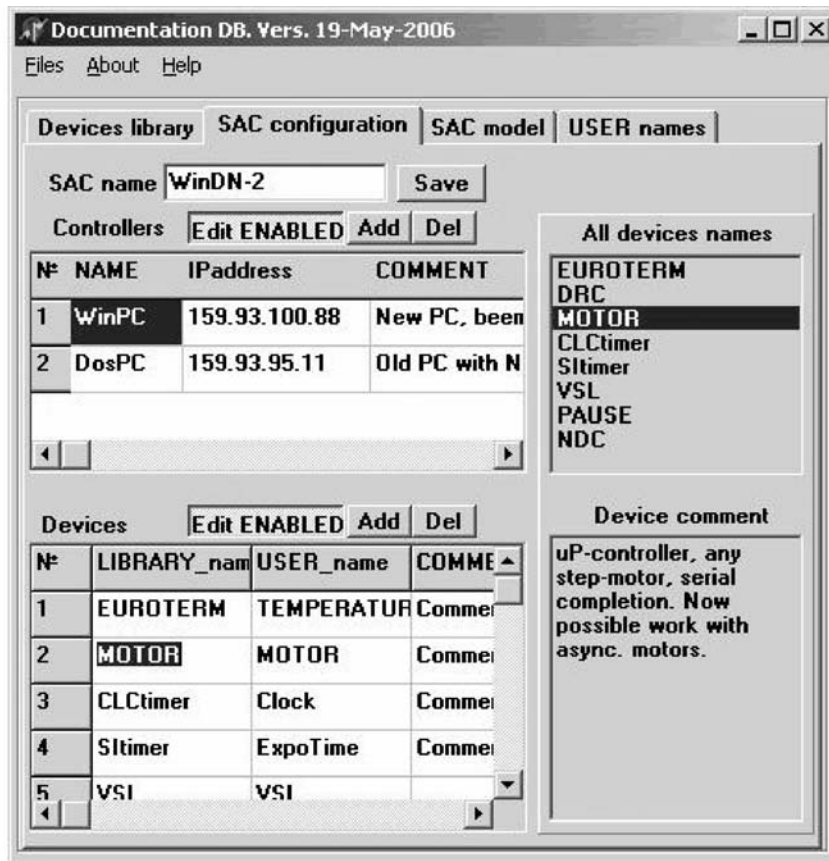


Рис. 3. Окно интерфейса пользователя, предназначенное для описания состава драйверного слоя САУ — максимального состава драйверов

Каждый вход **таблицы изменяемых параметров** содержит:

- два названия: заданное программистом и используемое пользователем (на начальном этапе совпадают);
- информацию об используемом варианте значения параметра (по умолчанию, фиксированное пользователем, изменяемое по прикладной методике);
- текущее значение параметра.

На рис. 4 представлено окно составления логической схемы САУ — состава контроллеров и способа подключения драйверов.

Таблица заданий содержит тексты заданий на выполнение сеансов работы (матрицы параметров) и комментарии пользователя к каждому заданию, заполняется по мере подготовки заданий пользователем и их исполнения.

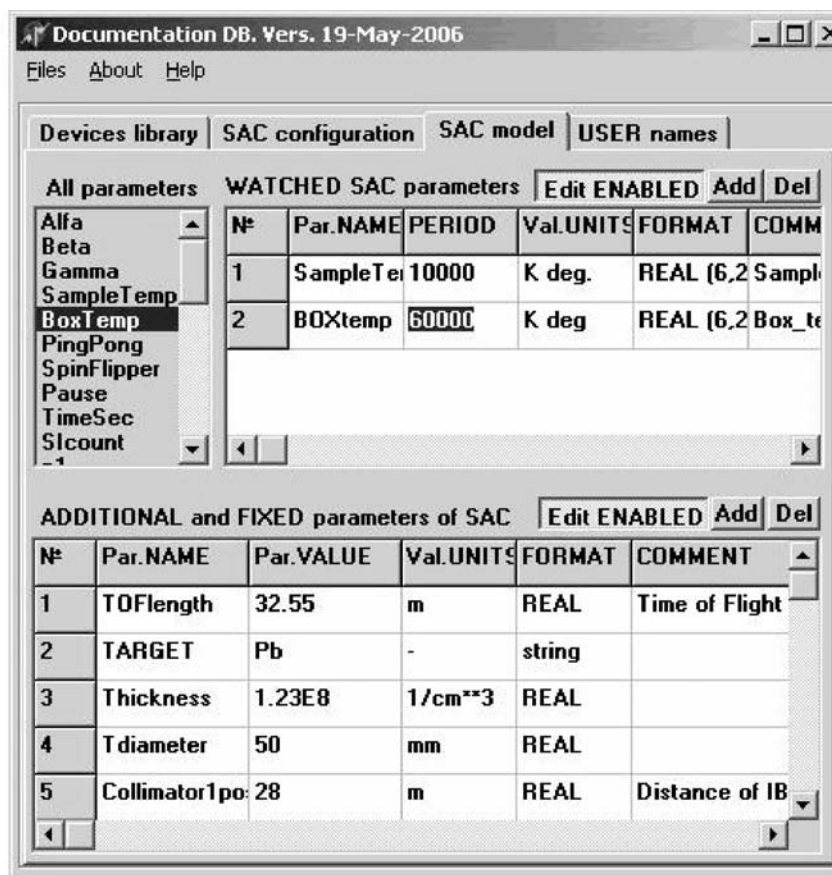


Рис. 4. Описание конфигурации контроллеров — логическая схема САУ

2.2. Программа подготовки задания. Процесс подготовки задания на сеанс работы состоит в том, что пользователь средствами программы **PSJ** составляет список условий, при которых должна быть выполнена регистрация данных и их сохранение в файлах. Такое задание программа автоматически представит в виде таблицы, столбцы которой соответствуют изменяемым параметрам, а строки содержат значения параметров, определяющих условия выполнения работы (векторы состояния). Число строк определяет количество файлов (точнее — наборов экспериментальных данных), которые будут записаны в процессе выполнения задания.

В программе **PSJ** объект управления представляется списком названий параметров:

- характеризующих условия работы, изменяемые с помощью программ управления специальным оборудованием (терморегуляторы, гониометры, источники тока и т. д.);
- содержащих справочную информацию об условиях измерения файлов, не используемую для управления оборудованием, но требующуюся при математической обработке данных (расстояние источник-детектор, тип детектора, назначение измерения — фон, пучок, мишень, и др.) или включаемую в отчеты;
- значения которых требуется наблюдать с заданной периодичностью; используется драйверами в специальном режиме, когда считываемые значения передаются исполняющей подсистеме (и программам обработки) без дополнительных команд, в режиме прерывания;
- несущих текстовые описания планируемых измерений.

Список таких параметров, составленный с помощью **PSJ**, является параметрической моделью САУ для конкретного сеанса работы.

Программа **PSJ** представляет собой специальный редактор, который может работать в нескольких режимах.

2.2.1. Режим составления задания с новым списком изменяемых параметров. По команде открывается окно с тремя полями списков:

- 1) названия всех параметров спектрометра (этот список всегда заполнен, т. к. параметрическая модель САУ задана уже при описании конфигурации);
- 2) названия параметров, изменяемых в планируемом сеансе работы;
- 3) поле списков значений выбранных изменяемых параметров.

Данное окно предоставляет возможность

- 1) в списке параметров объекта выбрать названия тех параметров, значения которых будут изменяться в данном сеансе, и занести их в список изменяемых параметров;
- 2) задать нужную очередность изменения значений параметров (последовательность в списке);
- 3) для каждого параметра задать список используемых в сеансе значений.

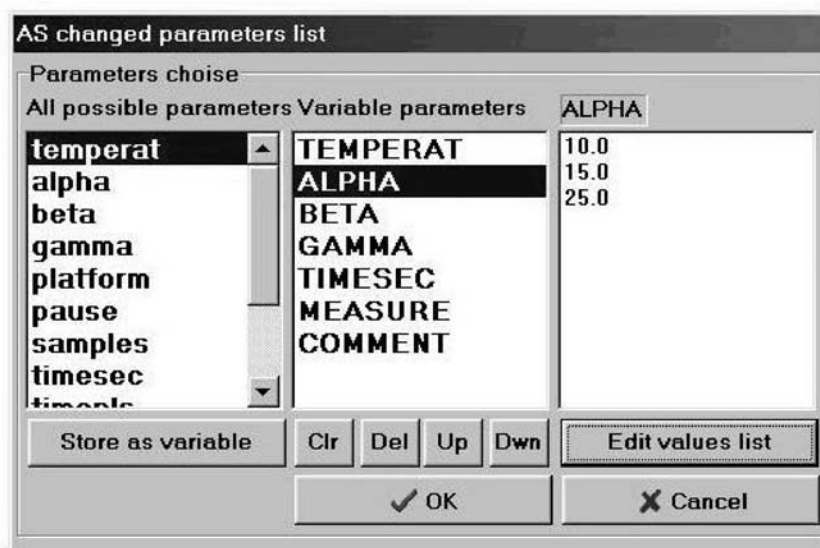


Рис. 5. Окно редактирования списка изменяемых параметров и списков их значений

На рис. 5 представлен вид окна редактирования списка изменяемых параметров и списков их значений.

После завершения процедуры редактирования программа автоматически создает таблицу состояний для заданных наборов значений параметров с учетом требующейся очередности их изменения. Вид такой таблицы представлен на рис. 6. На этом этапе, если требуется, имеется возможность дополнительного редактирования: убрать, добавить в таблицу строки и изменить содержание любого поля. Окончательно подготовленное задание по команде записывается в таблицу новых заданий пользователя.

2.2.2. Режим редактирования ранее подготовленного задания. Ранее подготовленное задание может быть прочитано редактором **PSJ**, представлено в виде таблицы и отредактировано. Результату редакции можно присвоить новое название и записать в качестве нового задания.

2.2.3. Помимо списка изменяемых параметров, есть возможность редактировать список параметров для протоколирования и список параметров, значения которых требуется наблюдать с заданной периодичностью.

2.2.4. В распоряжении программистов и пользователей САУ имеются средства редактирования значений настроечных (фиксируемых) параметров. Программа подготовки задания запоминает эти настройки.

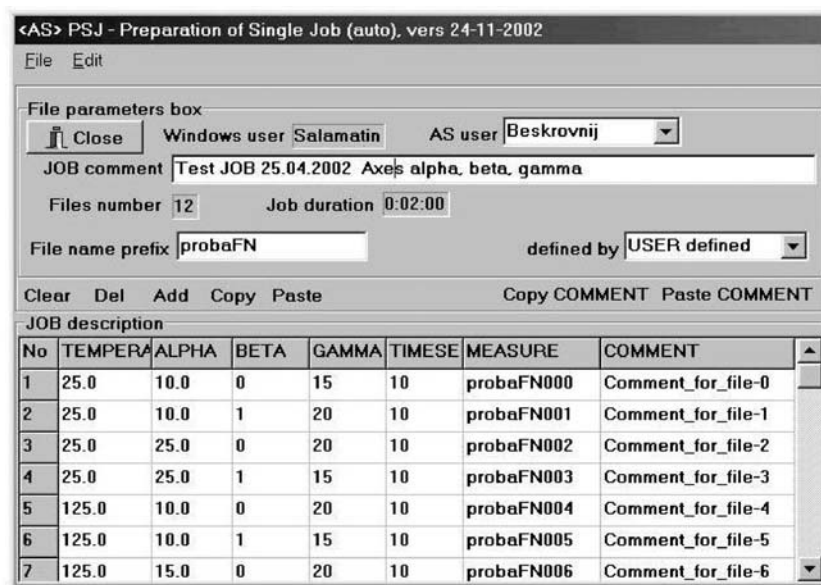


Рис. 6. Задание (матрица параметров), составленное программой **PSJ**

Программа **PSJ** может использоваться для коррекции задания в процессе его выполнения.

В отличие от работ [1, 2] данная реализация программы подготовки задания не зависит от конфигурации САУ и прикладной методики решения задачи, а также не требует знания языков программирования или специальной терминологии. Настройка на конкретное оборудование и задачу выполняется автоматически благодаря использованию параметрической модели САУ. **PSJ** может использоваться при решении широкого спектра задач: физические эксперименты, задачи мониторинга, задачи юстировки, поиска экстремумов, калибровки, комплексные проверки и тестирование САУ при любых сочетаниях варьируемых параметров (условий работы).

3. СТРУКТУРА ДРАЙВЕРОВ И СТРУКТУРА РАСПРЕДЕЛЕННОЙ САУ

По мере развития средств вычислительной техники и накопления опыта разработки происходит систематическое усложнение САУ. Сегодняшние требования к их структуре — распределенная конфигурация, возможность масштабирования, быстрая адаптация к изменениям методики работы. При этом особая роль отводится программам управления специальным оборудованием

(драйверам). Эти программы обычно занимают менее 10% в составе кода САУ, однако затраты на их разработку или адаптацию непропорционально велики. Известны работы [3–5], в которых вводятся специальные правила написания драйверов и способ интеграции их в систему. Общим ограничением этих работ является необходимость перетрансляции системы при изменении состава используемых драйверов или методики работы. В данной работе поставлена задача разработки такой методики включения в САУ новых драйверов, которую может использовать пользователь без применения среды программирования. При разработке **структуры драйвера** преследовались цели:

- разработка САУ в виде распределенной системы, масштабируемой силами пользователя;
- обеспечение преемственности и минимизация кода драйвера;
- автоматизация интеграции драйверов в систему.

Масштабирование включает возможность управлять количеством ЭВМ, которые САУ использует в процессе работы, и составом специального оборудования. Для реализации этих свойств потребовалось разработать специальную структуру драйверов, логическую структуру САУ и базовые модули комплекса, настраиваемые без применения среды программирования.

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

- 1) выполнение рутинных преобразований при получении соответствующей команды и вызов требуемой прикладной функции драйвера;
- 2) 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 наслоения кода, отражающего прикладную методику;
- 3) повышает надежность комплекса программ благодаря использованию многократно проверенного кода интерфейса.

Структура распределенной САУ представлена на рис. 7. По функциональному назначению в составе САУ мы выделяем: ЭВМ пульта оператора, сервер приложения, ЭВМ-контроллеры и временно подключающиеся ЭВМ наблюдателей. Возможное назначение ЭВМ определяется составом загруженных в нее базовых программ. Для реализации данной структуры разработаны базовые модули **Terminal**, **TCPclient**, **TCPserver**, **SetVec**, **FrontEnd**. Функции логических ЭВМ данного комплекса описываются ниже.

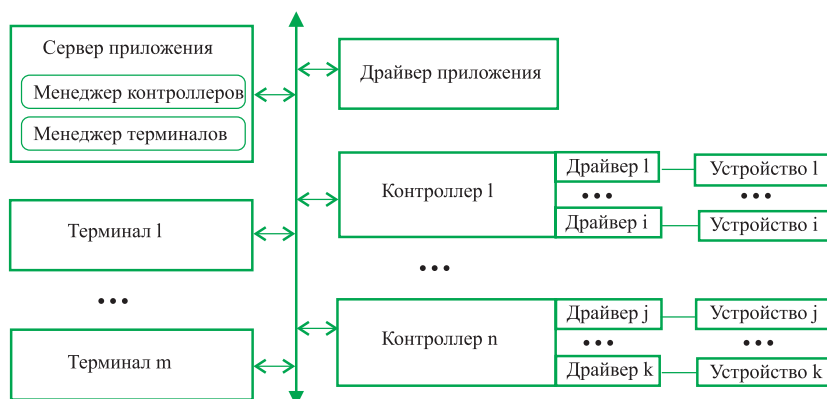


Рис. 7. Структура распределенной САУ

На **пульту оператора** работает программа **Terminal** — интерфейс пользователя, позволяющий управлять параметрами и режимом работы САУ;

Сервер приложения осуществляет:

- управление конфигурацией САУ — регистрирует подключающиеся Пульты операторов и контроллеры;
- управление конфигурацией контроллеров — составом драйверов на основании данных из документации САУ в БД;

- выполнение задания, представленного матрицей параметров;
- передачу команд драйверам и прием ответных сообщений (**TCPserver**).

Основная функция — выполнение задания — осуществляется модулем **SetVec**. Модуль выбирает из матрицы очередной вектор требуемого состояния объекта. На основании данных о предыдущем и требуемом состояниях программа **SetVec** выполняет следующие действия:

- 1) вычисляет вектор изменения состояния САУ;
- 2) составляет список команд управления отдельными узлами оборудования САУ;
- 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 САУ полностью в автоматическом режиме.

4. РЕЗУЛЬТАТЫ

Разработана база данных, обеспечивающая **информационную поддержку** САУ как при настройке ее базовых модулей служебными программами, так и на этапе выполнения работы. Разработанная программа **PSJ** подготовки задания инвариантна относительно изменений прикладной методики работы и конфигурации САУ и пригодна для **использования без изменений в различных САУ**. Использование рассмотренного способа подготовки и исполнения

задания дает **сокращение объема необходимого программирования** при одновременном увеличении реализуемых возможностей. Без дополнительного программирования реализуется

- возможность вмешательства в автоматическую работу САУ с целью изменить содержание выполняемого задания;
- автоматическое определение точки входа для продолжения работы САУ после сбоя питания и др. с известными и восстанавливаемыми потерями данных.

Фрагменты описанной системы подготовки задания использовались в течение ряда лет в ПО ряда спектрометров [6–8] на реакторе ИБР-2 [9].

Описанные в методы и алгоритмы использованы в программном обеспечении спектрометров СОСО, DN-2 [10] на реакторе ИБР-2. Эксплуатация САУ для спектрометра DN-2 [11–13] в 2003–2006 гг. 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 ТСР/ІР;
- использовать в САУ оборудование различных стандартов одновременно.

ЛИТЕРАТУРА

1. *Astakhova N. V. et al.* // Proceedings of the Second International Workshop on Data Acquisition Systems for Neutron Experimental Facilities (DANEF-2000), Dubna, 5–7 June 2000, JINR, Dubna, 2001, P. 275.
2. *Кирилов А. С. и др.* Сообщение ОИЯИ Р13-2003-66. Дубна, 2003.
3. <http://www.opcfoundation.org>
4. <http://www.labview.com>
5. <http://www.can-expo.ru/files/2006/>
6. *Вагов В. В. и др.* Сообщение ОИЯИ Р14-83-898. Дубна, 1983.
7. *Ostanevich Yu. M.* // Macromol. Chem. 1988. V. 15. P. 91.
8. *Балука Г. и др.* Препринт ОИЯИ Р10-12960. Дубна, 1980.

9. *Ананьев В. Д. и др. // ПТЭ. 1977. №5. С. 17.*
10. *Астахова Н. В. и др. // ПТЭ. 2005. №5. С. 36.*
11. *Мазный Н. Г. // Труды Всероссийской научно-технической конференции «Методы и средства обработки информации (МСО-2005)», 5–7 окт. 2005, МГУ; М.: Изд. отд. фак. ВМиК МГУ, 2005. С. 97.*
12. *Биккулова Н. Н. и др. // Изв. РАН. Сер. физ. 2006. Т. 70. С. 559.*
13. *Vasilovskiy S. G. et al. // Z. Kristallorg. Suppl. 2006. V. 23, P. 467.*

Получено 26 июня 2007 г.

Редактор *М. И. Зарубина*

Подписано в печать 12.10.2007.

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

Усл. печ. л. 1,12. Уч.-изд. л. 1,37. Тираж 305 экз. Заказ № 55920.

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

E-mail: publish@jinr.ru

www.jinr.ru/publish/