

P13-2013-86

К. М. Саламатин<sup>1,\*</sup>

**ВАРИАНТ ПОСТРОЕНИЯ  
КОМПОНЕНТНОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ  
ЭКСПЕРИМЕНТОВ ДЛЯ СПЕКТРОМЕТРИИ  
С ИСПОЛЬЗОВАНИЕМ СЕТЕВЫХ ТЕХНОЛОГИЙ**

Направлено в журнал «Программирование»

---

<sup>1</sup> Международный университет «Дубна»

\* E-mail: del@tmpk.ru

Саламатин К. М.

P13-2013-86

Вариант построения компонентной системы автоматизации экспериментов для спектрометрии с использованием сетевых технологий

На основе анализа современных сетевых технологий и специфики систем автоматизации экспериментов (САЭ) раскрыты причины и источники трудностей повторного использования компонентов САЭ. Сформулирована концепция компонентной САЭ, обеспечивающей возможность сопровождения и модификации САЭ пользователями в соответствии с изменением методики эксперимента. Определены ключевые задачи, решение которых необходимо для построения такой системы: разработка универсальной подсистемы описания методики эксперимента и распределенной среды обслуживания взаимодействия компонентов. Описано решение этих задач. Ряд разработанных типовых компонентов проверен в реальных экспериментах на реакторе ИБР-2. Полученные результаты повышают надежность и существенно сокращают время разработки и модификации САЭ.

Работа выполнена в соответствии с протоколом о совместных работах ЛНФ ОИЯИ и университета «Дубна» в области разработки методики программирования САЭ.

Препринт Объединенного института ядерных исследований. Дубна, 2013

Salamatin K. M.

P13-2013-86

A Variant of the Component-Based System for Spectrometry Experiments Automation Constructed Using Network Technologies

Based on the analysis of modern network technologies and specificity of experiment automation systems (EAS), the causes and sources of the difficulties arising under repeated usage of EAS components are shown. The concept of the component-based EAS which provides EAS modification in accordance with the change of the experiment method by the user is formulated. The key tasks necessary for the construction of such a system are identified: the development of a universal subsystem for the experiment method description and an environment for distributed components interaction. The implementation of these tasks is described. Some of the developed typical components are checked in real experiments at the IBR-2 reactor. These results increase reliability and significantly reduce the time of development and modification of the EAS.

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

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

Для систем автоматизации экспериментов (САЭ) характерно частое изменение методики исследования, что приводит к необходимости изменять состав используемого аппаратного и программного обеспечения. Из литературы известно, что на модификацию САЭ в лучшем случае тратится около шести месяцев работы коллектива, включающего нескольких программистов. Причины длительных сроков: комплексный характер работы, отсутствие адекватной стратегии повторного использования программного обеспечения, проблемы при попытке отторжения готовой САЭ от разработчиков и передачи функций ее модификации и сопровождения экспериментаторам, выражающиеся в повышенных требованиях к квалификации пользователя САЭ, необходимости освоения интерпретаторов, структуры конфигурационных файлов и др.

Статья посвящена методике снижения затрат на разработку и модификацию САЭ, повышения надежности систем и эффективности экспериментальных исследований. Основным критерий затрат — сроки разработки и модификации САЭ. Повторное использование — основная методология, которая применяется для сокращения трудозатрат при разработке сложных систем. Под повторным использованием понимается возможность использовать компоненты в различных экспериментах и системах без изменения самих компонентов или других составных частей САЭ. В работе поставлена задача: разработать архитектуру САЭ, методику ее компоновки и алгоритмы, обеспечивающие возможность многократного использования разработанного кода, а также передачи функции модификации системы пользователям.

Область интересов автора — автоматизация спектрометрии нейтронов на реакторе ИБР-2 [1] и ускорителе ИРЕН [2], поэтому рассуждения будут иллюстрироваться примерами из области спектрометрии. Однако решения выполнены в общем виде и пригодны для использования в другой проблемной области. Рассмотрим, что мешает повторному использованию компонентов.

## **1. ИСТОЧНИКИ И ПРИЧИНЫ ПОМЕХ ПОВТОРНОМУ ИСПОЛЬЗОВАНИЮ КОДА**

Основных источников помех повторному использованию компонентов два: 1) связанность компонентов и 2) изменчивость методики эксперимента.

Связывание компонентов может быть статическим и динамическим. При статическом связывании нужна для связывания информация (например,

сетевые адреса взаимодействующих компонентов) тем или иным способом фиксируются до начала работы САЭ. Статическое связывание обеспечивает наибольшую скорость выполнения взаимодействия, но при этом связь между компонентами системы становится более тесной, количество критических взаимозависимостей растет. В этом случае сбой одного из компонентов с большей вероятностью приводит к потере работоспособности всей системы, и даже небольшие изменения могут привести к необходимости привлечения сторонних специалистов (программистов) для выполнения дополнительного кодирования. Очевидно, для САЭ требуется использовать динамическое связывание, при котором необходимая для взаимодействия информация будет определена в процессе работы системы.

Изменчивость методики — неустранимая особенность исследовательской работы. Если новая методика требует использовать оборудование, не предусмотренное заранее в конструкции экспериментальной установки, это приведет также к изменению состава используемых компонентов и необходимости отредактировать ранее использовавшиеся компоненты. Как правило, при этом редактируется компонент, управляющий последовательностью выполнения операций в эксперименте, и сборка системы выполняется заново с использованием транслятора. Чаще всего для упрощения процесса модификации САЭ вводится упрощенный язык, интерпретатор включается в состав САЭ, и процедура эксперимента (скрипт) описывается на этом языке, а адресная информация помещается в конфигурационные файлы или передается через списки параметров, т. е. вводится статическая связанность.

Рассмотрим возможные пути решения этих проблем.

## 2. ВЫБОР СЕТЕВЫХ ТЕХНОЛОГИЙ ДЛЯ ИСПОЛЬЗОВАНИЯ В САЭ

Методы построения компонентных приложений наиболее широко представлены в сетевых технологиях. С целью сократить объем предстоящих собственных разработок и возможных ошибок был выполнен анализ ряда сетевых технологий и сделан выбор (с учетом специфики задач автоматизации экспериментов) оптимальных вариантов для использования в САЭ [3]. Рассмотрены методы RPC, технологии RMI, CORBA, DCOM, Ice и др., средства разметки сообщений XML, JSON, SOAP, технологии поиска компонентов, архитектура SOA и др.

Рассмотренные технологии имеют следующие свойства:

1. Для классического RPC (клиент/сервер) характерны *синхронный вызов*, способ взаимодействия *один-к-одному* и др. свойства, вносящие осложнения в системы реального времени.

2. RMI имеет жесткую *ориентацию на Java*.

3. DCOM возможно использовать *только в Windows*.

CORBA более полно отвечает требованиям, возникающим при реализации приложений с изменяемой логикой. Однако важнейшее свойство CORBA — универсальный механизм динамического связывания — для САЭ избыточно сложный [4]:

- требуется *включение специальных элементов в среду* обеспечения взаимодействия компонентов и в компоненты;
- реализация требует *нескольких сот операторов*;
- выполнение требует *ряда операций для настройки взаимодействия*, что в итоге в 40 раз медленнее, чем при статическом связывании;
- реализует схему взаимодействия *один-с-одним*.

Подробный анализ недостатков CORBA можно найти в статье [5]. Недавняя реинкарнация CORBA в Ice обеспечивает асинхронное взаимодействие, но еще более усложняет схему динамического связывания. В работе [6], посвященной анализу своего опыта разработки и эксплуатации Open Inspire, основанного на использовании JAVA и CORBA, дана отрицательная оценка результатов их использования. В данной работе принято решение выполнить разработку *механизма динамического связывания компонентов с учетом специфики решаемых задач*. Для разработки нового механизма связывания компонентов потребовались открытые сетевые технологии форматирования и сериализации сообщений и поиска компонентов. Сравнение характеристик JSON-RPC, XML-RPC и SOAP привело к решению использовать JSON-RPC. Для поддержки динамической компоновки САЭ использован протокол SLP — единственный, получивший статус RFC.

Оптимальным выбором архитектуры является использование SOA [7]. Эта архитектура позволяет решить проблему интеграции прикладной системы с возможностью автоматического изменения используемого состава компонентов в соответствии с эволюцией логики приложения (изменением методики эксперимента).

### **3. АНАЛИЗ И КЛАССИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ СОСТАВЛЯЮЩИХ САЭ В ОБЛАСТИ СПЕКТРОМЕТРИИ НЕЙТРОНОВ**

В опубликованных работах учет методики эксперимента и связывание компонентов осуществляется разными способами — в фрагментах кода системы, скриптами, средствами компонентных технологий RMI, DCOM, CORBA и др. Эти решения вносят связанность компонентов, в итоге попытка реализовать не учтенную ранее методику эксперимента приводит к изменениям готовых программ, избыточному коду в компонентах, замедлению работы, что ухудшает возможность повторного использования программ. *В литературе не были найдены средства описания новой (не учтенной при*

разработке САЭ) методики, не требующие изменения кода, скриптов или других частей, входящих в состав САЭ. Чтобы получить информацию для разработки нового алгоритма динамического связывания компонентов и подсистемы описания методики эксперимента с нужными свойствами, был выполнен анализ функциональных компонентов САЭ.

САЭ должна решать следующие задачи:

- 1) управления окружением образца в соответствии с методикой эксперимента;
- 2) регистрации, представления данных в формате, удобном для обработки, и сохранения потока данных от детекторов;
- 3) контроля корректности работы экспериментальной установки и достоверности регистрируемых данных.

Необходимый шаг на пути обеспечения *возможности повторного использования* — разделение кода САЭ на отдельные функционально законченные модули, назначение которых, например, следующее:

1. Интерфейс пользователя (возможно несколько).
2. Подсистема регистрации данных DAQ (ввод, преобразование и архивирование).
3. Управление устройствами окружения образца, определяющими условия регистрации данных.
4. Управление экспериментом.
5. Мониторинг состояния объекта.
6. Обработка отказов.
7. Предварительная обработка данных.
8. Визуализация данных и др.

Анализ состава функциональных компонентов САЭ [8] позволяет сделать следующие выводы:

- Требования к скорости обмена данными являются критическими только для компонентов DAQ, и эта задача решается посредством организации общего доступа к оперативной памяти, дополнительно возможно использовать специальную внешнюю память.

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

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

- В составе САЭ присутствуют другие функциональные пары — источники информации, вырабатываемой в процессе работы САЭ, и потребители

этой информации. Источник может публиковать информацию различного типа (адреса экспериментальных данных, информацию о состоянии узлов установки, диагностические сообщения и др.). Потребителю требуется информация определенного типа, потребителей информации любого типа может быть несколько либо ни одного. Методика эксперимента не определяет схему взаимодействия источников и потребителей информации.

- Можно сформулировать *две группы* компонентов САЭ:

1) *базовые компоненты* с детерминированным характером взаимодействия, выполняющие основные задачи эксперимента; характер взаимодействия — один-к-одному;

2) компоненты (и методы базовых компонентов), реализующие *вспомогательную логику* — сервисные функции, обработку нештатных ситуаций и др., не влияющие (в штатных условиях) на выполнение основных задач; характер взаимодействия — один-ко-многим.

- В группе базовых компонентов присутствует *только два типа клиентов*, в качестве результата им требуются только подтверждение выполнения заказанного действия для синхронизации работы. Это — интерфейсы пользователя и программа управления экспериментом. По сути *это* — *управляющие программы*, управление — однонаправленное. Реализация обратной связи относится к вспомогательной логике.

- В группе базовых компонентов присутствует *только два типа сервисов*, выполняющих *основную работу*, определяемую методикой эксперимента. Это подсистемы регистрации данных (DAQ) и компоненты управления устройствами окружения образца.

#### **4. ПРЕДЛАГАЕМАЯ КОНЦЕПЦИЯ И КЛЮЧЕВЫЕ ЗАДАЧИ ПРИ ЕЕ РЕАЛИЗАЦИИ**

Комплексный подход — учет специфики экспериментальных исследований, использование современных технологий и разработка дополнительных алгоритмов и методов — позволяет построить систему, komponуемую из типовых программных модулей, концепция которой включает следующие положения:

- система собирается из взаимодействующих типовых *функционально законченных компонентов*, предоставляющих некоторый набор методов; компоненты представляются в исполняемом формате, для них *определен внешний интерфейс* — способ удаленного вызова методов и ответная реакция;

- *код компонентов системы инвариантен* относительно изменений методики исследования;

- методика эксперимента описывается с помощью *унифицированной диалоговой подсистемы составления задания*, инвариантной относительно методики эксперимента и конструкции спектрометра;

- *автоматическая компоновка* системы в соответствии заданием на эксперимент;
- *перенос компонентов на другие ЭВМ* в пределах локальной сети не должен разрушать систему, требовать изменения описания ее конфигурации или перекомпиляции любых составляющих САЭ;
- процессом выполнения базовой логики управляет *типовая управляющая программа*, алгоритм которой инвариантен относительно методики эксперимента;
- эксперимент выполняется в *полностью автоматическом режиме* с возможностью перейти в диалоговый;
- возможна передача функций сопровождения и *модификации САЭ пользователям*.

На основании анализа сетевых технологий разработки компонентных приложений и особенностей САЭ сделан вывод о необходимости решения двух ключевых задач, это:

- 1) разработка универсального механизма описания методики эксперимента, не требующего изменения кода готовых компонентов при изменении методики;
- 2) разработка среды взаимодействия компонентов, адаптированной к специфике САЭ и инвариантной к изменениям методики эксперимента и вспомогательной логики.

Разработка интерфейсов пользователя, программы управления экспериментом, компонентов управления окружением образца, DAQ и др. не содержит идеологических проблем, ряд вариантов таких модулей проходит испытания в условиях реальных экспериментов [9]. Рассмотрим варианты решения ключевых задач.

## **5. ПОДСИСТЕМА ОПИСАНИЯ МЕТОДИКИ ЭКСПЕРИМЕНТА**

Для описания методики и составления задания на эксперимент требуется информация об устройствах и работающих с ними компонентах (паспорта устройств). Рассмотрены два альтернативных алгоритма получения этой информации:

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



Алгоритм составления списков активных компонентов по их типам присутствует в механизме обслуживания взаимодействия компонентов (описан ниже), с его помощью выполняется поиск:

- интерфейсом пользователя — программы управления экспериментом;
- программой управления экспериментом — компонентов управления окружением образца и компонентов DAQ;
- подсистемой мониторинга — нужных компонентов системы и обнаружение возможных коллизий (например — несколько управляющих программ);
- поиск Менеджера событий и др.

Однако для реализации подсистемы описания методики эксперимента и формирования задания на эксперимент выбран первый вариант по следующим причинам:

- задание на эксперимент возможно подготовить заранее на любой ЭВМ без запуска САЭ;
- можно подготовить к выполнению пакет заданий;
- вариант лучше защищен от ошибок пользователя.

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

**5.1. Программа составления паспортов** устройств создает и заносит в БД описание устройства в формате JSON. Паспорт содержит:

- версию паспорта;
- имя контроллера устройств;
- тип компонента;
- уникальный ID (GUID), используемый для адресации средствами коммуникаций;
- перечень параметров устройства. Параметр имеет имя, тип значения, значение по умолчанию и диапазон допустимых значений.

Состав параметров (прикладной протокол) определяется разработчиком компонента.

**5.2. Программа составления задания** использует список доступных устройств (и компонентов) из БД. Из этого списка пользователь в диалоге выбирает нужные в данном эксперименте компоненты и составляет список устройств, для которых в процессе эксперимента последовательно будет задаваться несколько состояний в соответствии с содержанием задания. Выбирая названия устройств из этого списка, пользователь может задать для каждого список значений управляемого параметра. На основании этих данных PSJ создает таблицу, в которой пользователю предоставляется дополнительная возможность изменить состав и последовательность работы компонентов

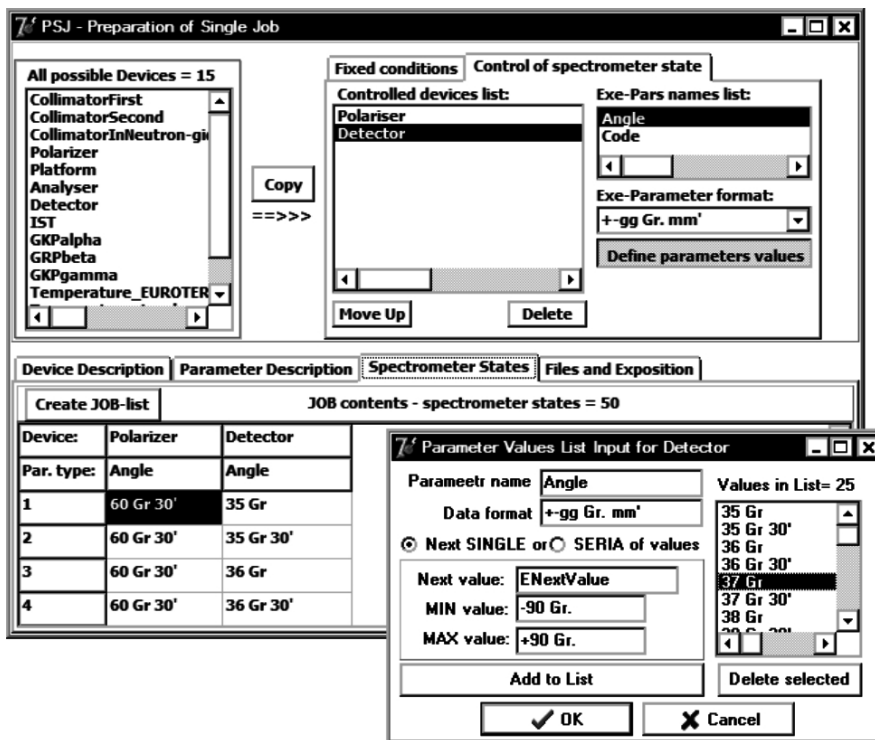


Рис. 1. Вид окна программы PSJ в режиме редактирования

управления оборудованием, а также состав используемых значений параметров. Вид окна программы PSJ в режиме редактирования показан на рис. 1. Результат работы PSJ — файл с табличным описанием конечного автомата, реализующего нужные в эксперименте состояния аппаратной системы. В каждом состоянии выполняется экспозиция данных. Структура файла задания на эксперимент показана на рис. 2.

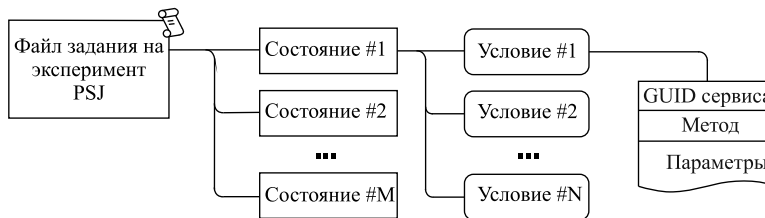


Рис. 2. Структура файла задания на эксперимент

При описании методики эксперимента пользователем применяется только терминология его проблемной области — названия узлов спектрометра, угловые положения и др. Информация для поиска сервиса (GUID) и вызова метода переносится из паспорта устройства (документации) в файл задания автоматически.

## **6. РАСПРЕДЕЛЕННАЯ СРЕДА ОБЕСПЕЧЕНИЯ ВЗАИМОДЕЙСТВИЯ КОМПОНЕНТОВ**

Среда обеспечения взаимодействия компонентов — важнейший компонент САЭ. Средствам обеспечения взаимодействия процессов выдвигаются следующие требования:

- 1) автоматический поиск и динамическое связывание компонентов;
- 2) асинхронный механизм вызова методов, так как при синхронном вызове снижается скорость выполнения логики приложения (и пропускная способность), а также усложняется программирование системы реального времени, где процессы выполняются одновременно и события, требующие обработки, возникают асинхронно;
- 3) возможность передать информацию нескольким другим процессам;
- 4) одна и та же среда взаимодействия должна обрабатывать все обмены в рамках САЭ, так как гомогенную систему намного легче программировать и поддерживать;
- 5) прозрачность взаимодействий в распределенной системе; где бы ни исполнялся процесс, он должен иметь возможность взаимодействовать с любым процессом в системе, используя единый механизм, который не зависит от размещения процессов;
- 6) возможность перемещения процесса с одной машины на другую; это облегчит устранение аварийных ситуаций, возникающих при выходе из строя ЭВМ;
- 7) интерфейс не должен зависеть от границ ЭВМ — код должен быть одинаков при обращении к процессу на той же машине или на другой, в том числе и другого типа; 8) потеря процесса, ЭВМ или разрыв сетевой связи не должна приводить к разрушению остальной части системы;
- 9) автоматическая адаптация к используемой конфигурации.

В соответствии с этими требованиями разработан компонент DiCME (*Distributed Components Messaging Environment*). Среда взаимодействия является связующим звеном для трех слоев программ:

- 1) программы управления;
- 2) программ, выполняющих работу в рамках базовой логики;
- 3) программ вспомогательной логики.

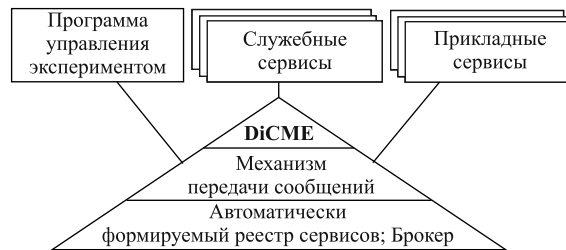


Рис. 3. Архитектура САЭ

DiCME обеспечивает базовую поддержку сервис-ориентированной архитектуры САЭ (рис. 3). В DiCME, помимо механизма передачи сообщений, включены функции формирования реестра компонентов, брокер сообщений и некоторые другие службы. Для обеспечения автоматического формирования реестра сервисов каждый компонент, реализующий какую-либо сервисную функцию, на этапе инициализации своего интерфейса к DiCME активирует маяк. Маяк периодически посредством multicast-сообщения информирует другие компоненты о своей доступности — сообщает свой идентификатор (GUID) и тип.

Одна из решающих характеристик среды взаимодействия — способ связывания компонентов. Для позднего (динамического) связывания информацию дает файл задания, в который PSJ автоматически занесла GUID и тип устройства, определенные в паспорте (из БД). В настоящее время используются два типа устройств — DAQ и CONDITION. Метод передачи (SendMSG) фиксирован, этот метод обеспечивает передачу сообщения исполняющему компоненту.

На рис. 4 показан алгоритм работы программы управления экспериментом (ПУЭ) и способ использования DiCME при реализации базовой логики САЭ. На вход управляющей программы поступает файл задания. ПУЭ выбирает описание очередного состояния системы (список условий) и передает описание каждого условия DiCME. Описание условия несет информацию о сервисе (GUID, тип), достаточную для его поиска и связывания с ним всегда одного и того же компонента — программы управления экспериментом. DiCME находит нужный компонент типа CONDITION и передает ему описание условия, содержащее список параметров. Список параметров — часть сообщения, интерпретируемая кодом компонента. Например, для управления положением поляризатора этот список в формате JSON имеет вид

```
{device:"polarizer", parameter:"angle", value:"30grad"}
```

Программа управления и DiCME прозрачны для списка параметров. Каждый компонент типа CONDITION должен вернуть ПУЭ сигнал

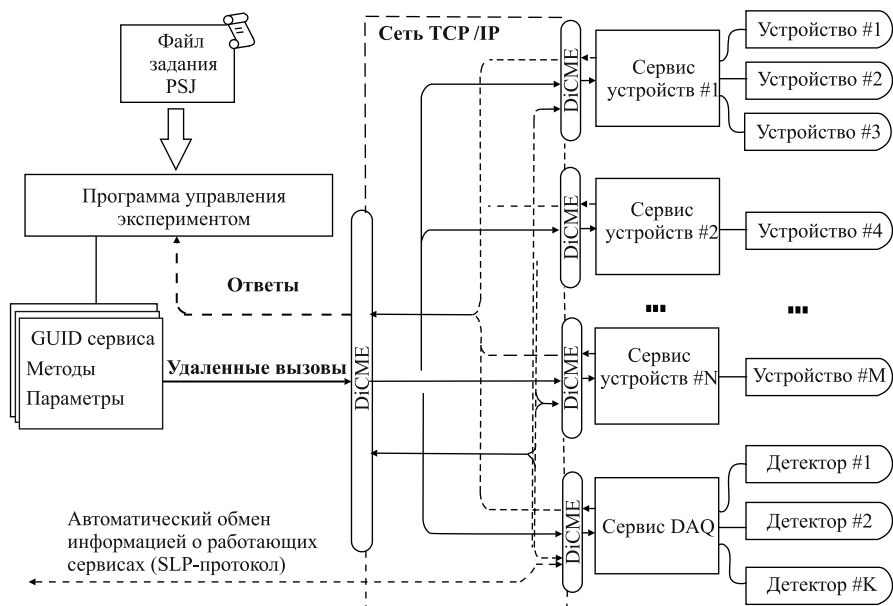


Рис. 4. Алгоритм работы управляющей программы и способ использования DiCME при реализации базовой логики САЭ

(DONE/ERROR) о завершении работы. После получения таких сигналов от всех устройств, перечисленных в описании состояния САЭ, ПУЭ включает регистрацию данных подсистемой DAQ. Сигнал о завершении экспозиции данных разрешает ПУЭ перейти к обработке описания следующего состояния в файле задания.

Назовем этот метод связывания «внешним», поскольку необходимая для связывания и параметризации действия информация вырабатывается средствами, внешними по отношению к ПУЭ компонентам и среде обеспечения взаимодействия. В отличие от CORBA, внешнее связывание полностью устраняет необходимость подготовительного диалога между компонентами и, благодаря автоматическому составлению в DiCME реестра компонентов, схема связывания эквивалентна по скорости статическому связыванию. Данный метод связывания не ограничивает развитие методики эксперимента, представленной в виде описания конечного автомата (списка состояний САЭ), и любые ее изменения не затрагивают среду взаимодействия.

Наиболее существенным отличием способа реализации вспомогательной логики от базовой является необходимость передать информацию нескольким процессам, состав которых, вообще говоря, источнику информации не известен.

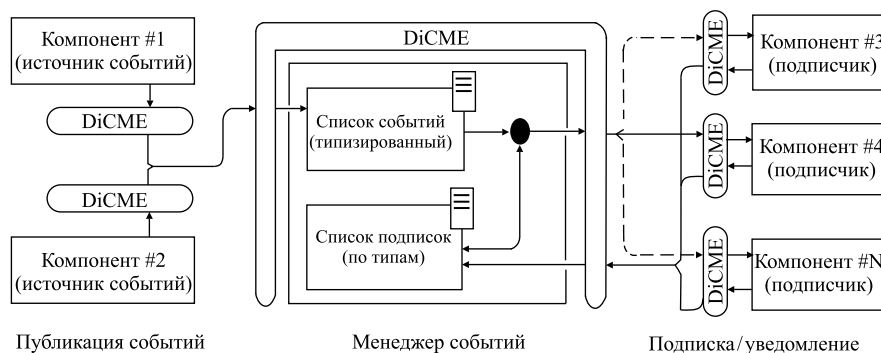


Рис. 5. Алгоритм работы DiCME при обслуживании вспомогательной логики

Для связывания вспомогательных компонентов выбран вариант алгоритма «подписки», при котором *потребитель однократно декларирует интерес к информации определенного типа*, после чего специальный сервис (Менеджер событий) обслуживает всех «подписавшихся» потребителей при появлении этой информации. На рис. 5 показан алгоритм работы DiCME при обслуживании вспомогательной логики САЭ.

В итоге наличие программы подготовки задания PSJ и реализация базовой и вспомогательной логики через DiCME с автоматическим поиском компонентов предоставляют возможность пользователю:

- управлять методикой эксперимента,
- динамически компоновать систему с нужными вспомогательными функциями просто путем запуска на любой ЭВМ локальной сети нужного сервиса.

## ЗАКЛЮЧЕНИЕ

В заключение приведем некоторые отличительные особенности данной работы:

- выявлены источники и причины помех к повторному использованию компонентов; разработаны методика и алгоритмы, которые позволили снять возникающие ограничения;
- разработан и применен *новый метод динамического связывания компонентов — внешнее связывание*. Благодаря внешнему связыванию не требуется изменение среды взаимодействия при изменении базовой и вспомогательной логики приложения; скоростные характеристики разработанного метода не уступают возможным при статическом связывании;

- разработаны *новый метод и алгоритмы описания методики* эксперимента; разработана и применена универсальная подсистема описания методики для экспериментов в области спектрометрии;

- введена *классификация компонентов* по принадлежности к базовой и вспомогательной логике эксперимента; выявлены и использованы при разработке алгоритмов особенности выполнения базовой и вспомогательной логики;

- использованы *различные механизмы обслуживания* базовой и вспомогательной логики;

- *поиск сетевого хранилища данных (NAS — Network Archive Service)* реализуется с помощью тех же механизмов, что и поиск других компонентов;

- разработанная среда взаимодействия позволяет *динамически изменять состав и адреса* базовых и вспомогательных компонентов, что облегчает восстановление работоспособности системы при отказах, улучшает условия эксплуатации и управления;

- сняты ограничения на развитие базовой и вспомогательной логики эксперимента *без дополнительного программирования*;

- *компоновка САЭ выполняется автоматически*; это позволяет выполнять модификацию САЭ для новых экспериментов силами пользователя без применения дополнительного программирования, составления скриптов или конфигурационных файлов;

- реализован метод *инкрементного развития САЭ*, при котором функциональные возможности САЭ можно наращивать путем включения новых компонентов *без изменения остальных частей системы*;

- разработанная *среда коммуникаций инвариантна* относительно
  - изменения базовой и вспомогательной логики приложения,
  - появления источников сообщений нового типа,
  - появления новых обработчиков событий.

Функции среды обслуживания взаимодействия процессов написаны в общем виде, не связаны с конкретной областью приложения и могут быть использованы в различных системах автоматизации экспериментов и системах автоматизированного управления в других проблемных областях.

Ряд компонентов проверен в экспериментах по исследованию длины рассеяния нейтронов на благородных газах [10], цикле экспериментов с поляризованными мишенями и ядрами на установке КОЛХИДА [11], экспериментах с источником нейтронов на базе мезонной фабрики в Красной Пахре [12] и др.

Работа выполнена в соответствии с протоколом о совместных работах ЛНФ ОИЯИ и университета «Дубна» в области разработки методики программирования САЭ.

## ЛИТЕРАТУРА

1. Драгунов Ю. Г. и др. Модернизация импульсного исследовательского реактора ИБР-2 // АЭ. 2012. Т. 113, вып. 1. С. 29–34.
2. Belikov O. V. et al. Physical Startup of the First Stage of IREN Facility // Proc. of Intern. Seminar «ISINN-17» (Dubna, May 27–29, 2009). Dubna: JINR, 2010. P. 10–16.
3. Саламатин К. М. Выбор технологии построения компонентной системы для автоматизации экспериментов в области спектроскопии нейтронов. Препринт ОИЯИ P13-2013-72. Дубна, 2013.
4. <http://kunegin.com/ref3/corba4/7.htm>; <http://www.rsdn.ru/article/corba/vsCORBA.xml>
5. Michi Henning. The Rise and Fall of CORBA // ACM Queue. V. 4, No. 5, June 2006; [http://citforum.ru/SE/middleware/corba\\_history](http://citforum.ru/SE/middleware/corba_history)
6. Flemming S. A. et al. The Open Inspire Architecture for Control, Data Reduction and Analysis // NOBUGS2008, Paper 134, 2008.
7. Дубова Н. На пути к SOA // Директор информационной службы. 2005. № 8. С. 12.
8. Саламатин И. М., Саламатин К. М. Разработка компонентной САЭ для физики низких энергий на основе использования сетевых технологий. Препринт ОИЯИ P13-2013-74. Дубна, 2013.
9. Швецов В. Н. и др. 8-входная система TOF для нейтронно-ядерных исследований по методу времени пролета // ПТЭ. 2012. № 5. С. 54–61.
10. Enik T. L. et al. Preparation for Testing Experiment on IREN Neutron Beam // Proc. of Intern. Seminar «ISINN-17» (Dubna, May 27–29, 2009). Dubna: JINR, 2010. P. 162–164.
11. Абов Ю. Г. и др. Установка КОЛХИДА для экспериментальных исследований взаимодействий поляризованных нейтронов с поляризованными ядрами. Препринт ОИЯИ P13-2008-69. Дубна, 2008.
12. Kuznetsov V. L., Kuznetsova E. V. Study of the Effects of Parity Violation in Neutron Diffraction in Perfect Single Crystal // Proc. of Intern. Seminar «ISINN-20» (Dubna, May 27–29, 2012). Dubna: JINR, 2013. P. 162–164.

Получено 13 августа 2013 г.



Редактор *М. И. Зарубина*

Подписано в печать 25.11.2013.

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

Усл. печ. л. 1,06. Уч.-изд. л. 1,29. Тираж 245 экз. Заказ № 58128.

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

E-mail: [publish@jinr.ru](mailto:publish@jinr.ru)

[www.jinr.ru/publish/](http://www.jinr.ru/publish/)