



Объединенный  
институт  
ядерных  
исследований  
дубна

P10-85-281

Г.Балука, И.М.Саламатин

ДИНАМИЧЕСКИ ФОРМИРУЕМАЯ  
ПРОГРАММНАЯ СИСТЕМА  
АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТОВ

Направлено на XIX школу по автоматизации  
научных исследований /Новосибирск, 1985/

1985

В работе<sup>/1/</sup> для создания систем автоматизации экспериментов /САЭ/ предложена методика динамического формирования конфигурации /ДФК/ программного обеспечения САЭ. Эта методика включает: представление структуры управляющей программы в виде иерархии виртуальных программных процессоров; динамическое формирование информационной базы САЭ; динамическое формирование состава и связей функциональных компонентов САЭ; модульную организацию функционального программного обеспечения, в том числе программного обеспечения оборудования КАМАК, и динамическую настройку таких программ в соответствии с конкретной конфигурацией оборудования КАМАК.

В данной работе описывается комплекс средств, реализующих эту методику.

## 1. ТРЕБОВАНИЯ К САЭ И ЕЕ НАЗНАЧЕНИЕ

Сложившаяся практика использования ЭВМ в экспериментах на реакторах ИБР-30, ИБР-2 имеет определенные особенности, в частности: продолжительность непрерывной работы ЭВМ в эксперименте может составлять от десятков минут до нескольких недель; степень автоматизации требуется высокая, вплоть до автоматического выполнения очереди экспериментов<sup>/2/</sup>; диапазон требований к объему программного сервиса широк, растет, и иногда целесообразно выполнение полной математической обработки данных на ЭВМ нижнего уровня<sup>/3/</sup>. Следствием этих условий является требование повышенной надежности программного обеспечения /ПО/ САЭ с целью получения результатов в условиях возможного дрейфа характеристик экспериментальной установки, сбоев и аварий оборудования и гибкости ПО САЭ по отношению к изменениям методики эксперимента.

Обеспечение надежности рассматривалось в двух планах:

1. Обеспечение надежности экспериментальных данных. Это реализовано путем разработки специальных функциональных программных модулей, выполняющих по заданным критериям анализ качества данных, контролирующих характеристики оборудования, подготавливающих и осуществляющих варианты автоматической коррекции работы САЭ.

2. Обеспечение устойчивости САЭ по отношению к сбоям оборудования, питания и некоторым другим авариям. Это реализовано комплексом мер, итогом которых явилась типовая процедура обеспечения

продолжения работы САЭ после сбоя, устранения аварии, восстановления характеристик установки /процедура рестарта/, оптимизирования по критерию минимальной потери экспериментальных данных.

Для описания задачи, решаемой ПО САЭ, введем ряд понятий.

Управляющий тракт - программа, используемая для изменения состояния узлов экспериментальной установки, программ, данных и др. Эти программы могут, например, выполнять пуск или остановку работы оборудования, индикацию данных или состояний, управлять системами замены, перемещения, ориентации исследуемых мишеней и др.

Регистрирующий тракт - программа, используемая для регистрации экспериментальных данных, информирования об изменении состояния физической установки или ЭВМ, программ, данных и др. Такие программы могут работать с регистрами данных аналого-цифровых преобразователей, счетчиков, блоками памяти и др.

Образец - такая совокупность управляющих и регистрирующих трактов, в которой тракты управления обеспечивают фиксацию экспериментальной установки в любом используемом состоянии на время экспозиции и синхронизацию работы набора регистрирующих трактов.

Пакет - совокупность образцов, определяющая используемые в данном эксперименте состояния экспериментальной установки и виды экспериментальных данных.

Эксперимент - последовательность измерений пакета образцов. Время эксперимента разделено на интервалы времени экспозиции образцов с целью ослабить влияние дрейфа характеристик установки на результаты эксперимента, а также для того, чтобы включить операции контроля характеристик установки и запоминания состояния системы для рестарта. Оптимальное время экспозиции  $t_э$  может быть найдено из соотношения  $t_э = (t_{RT})^{1/2}$ , где  $t_{RT}$  - время контроля и запоминания состояния системы,  $T$  - среднее время между потерями результатов из-за аварии, сбоя и т.д.

Сеанс - автоматически выполняемая последовательность экспериментов. В эту последовательность могут входить эксперименты на различном составе оборудования, подключенного к данной ЭВМ, выполняемые разными экспериментаторами.

По данной терминологии назначение программного обеспечения САЭ - управление сеансом работы ЭВМ. Такой процесс расчленен на ряд более простых, отдельными процессами управляют программные процессоры: сеанса, эксперимента, пакета и т.д. Назначение этих процессоров определено, а состав программных модулей, выполняющих конкретную работу, может меняться в процессе работы, поэтому мы будем говорить о виртуальных процессорах.

Назовем информационной моделью /ИМ/ процесса /или соответствующего ему процессора/ совокупность переменных, значения которых достаточны для описания текущего состояния управляемого процесса. При разработке достаточно сложная система обычно разде-

ляется на программные модули. Совокупность ИМ программных модулей, выполняющих процесс, входит в ИМ этого процесса. Эксперимент может быть представлен в виде комбинации непрерывных и дискретных процессов. ИМ дискретных процессов строится проще. В данной системе ИМ строится для дискретных состояний сеанса, детализируется до уровня образца, используется для запоминания состояния сеанса на внешних носителях и восстанавливается при рестарте сеанса.

## 2. ОБЩЕЕ ОПИСАНИЕ КОМПЛЕКСА СРЕДСТВ СОЗДАНИЯ САЭ

Комплекс средств создания программного обеспечения САЭ включает системы программирования общего назначения, предоставляемые штатной операционной системой RT-11, и специально разработанные средства.

Штатные средства ОС RT-11 используются для создания компонентов САЭ.

Структура специальных разработанных средств соответствует структуре проблемно-ориентированных модульных систем /ПОМС//4/ и включает:

- системное наполнение;
- функциональное наполнение, отражающее проблемную область;
- резидентную часть, управляющую выполнением сеанса работы ЭВМ.

2.1. Системное наполнение ПОМС включает программы составления заданий и системные программные модули, обслуживающие выполнение заданий.

Программы составления заданий используются разработчиками или пользователями, чтобы в режиме диалога описать:

- конфигурацию /состав, способ размещения в крейтах, способ подключения крейтов к ЭВМ/ экспериментального оборудования САЭ в стандарте КАМАК;
- состав функционального наполнения САЭ, используемого в автоматическом режиме выполнения эксперимента;
- задание САЭ на выполнение эксперимента или очереди экспериментов.

Описания конфигурации оборудования КАМАК и состава функционального наполнения САЭ составляют спецификацию САЭ в рамках данной ПОМС и используются системными модулями в процессе выполнения задания.

Программа составления задания на выполнение эксперимента обращается к специальной подпрограмме, запрашивающей у пользователя коды экспонируемых состояний экспериментальной установки, продолжительность экспозиции, названия файлов результатов, критерии контроля характеристик экспериментальных данных и некоторые другие значения. Эта специальная подпрограмма разрабатыва-

ется /или модифицируется/ индивидуально для каждой экспериментальной установки и ориентирована на пользователя.

Системные программные модули ПОМС выполняют ряд функций, в числе которых: динамическое формирование информационной базы САЭ; динамическое формирование в оперативной памяти состава функциональных программ, необходимых на данной фазе работы САЭ; коррекция описания текущего состояния эксперимента и сеанса для обеспечения возможности рестарта; рестарт при сбоях и авариях; занесение записей в файл журнала экспериментатора; занесение записей в файл протокола и др.

Системное наполнение ПОМС не зависит от методики эксперимента. Совокупность спецификации САЭ и заданий экспериментов является заданием ПОМС на автоматическое или автоматизированное выполнение сеанса работы ЭВМ на линии с экспериментальными установками.

2.2. Функциональное наполнение представлено расширяемым набором программных модулей, хранящихся в специальной библиотеке/5/ или в виде файлов ОС RT-11. Все программные модули разработанной ПОМС имеют следующие характеристики:

- структура модулей совпадает со структурой процедур на языке Паскаль; на список параметров, передаваемых модулю, не наложено ограничений, дополнительных к сформулированным в описании этого языка;

- все модули независимо транслируются, независимо обрабатываются редактором связей и представляются в перемещаемом формате загрузки .REL; для использования одного и того же модуля в различных САЭ не требуется повторять трансляцию или редактирование связей;

- модули, работающие с оборудованием КАМАК, инвариантны к изменению позиций блоков в крейтах и способа коммутации контроллера крейта/6/;

- модули, обрабатывающие внешние запросы прерывания, также имеют структуру процедуры; допускается список параметров в процедурах обработки прерываний/6/;

- модули могут обращаться к подпрограммам резидентной общей библиотеки подпрограмм;

- модули могут быть написаны на языках Паскаль, Макро-11, Фортран-4; возможно и планируется использование других языков программирования.

Основу функционального наполнения составляют программные модули, обслуживающие типичные тракты регистрации и управления. Узкоспециализированные модули обычно содержат обращения к группе типичных модулей.

2.3. Резидентная часть системы включает управляющую программу, перемещающий загрузчик функциональных программных модулей, программу динамического распределения памяти и резидентную библиотеку общих подпрограмм.

Начальное состояние резидентной части системы перед любым экспериментом одно и то же. Основной режим - продолжение ранее прерванной работы. Управляющая программа читает файл описания текущего состояния эксперимента и файл задания, формирует информационную модель эксперимента, загружает в оперативную память те функциональные программные модули, которые требуются в соответствии с алгоритмом работы на данной фазе эксперимента, и продолжает работу.

В рамках данной ПОМС создание САЭ начинается с создания файла спецификации САЭ для конкретного эксперимента. Предварительно должны быть разработаны недостающие программные модули функционального наполнения ПОМС. Выполнение работы /эксперимента, сеанса/ САЭ начинается с чтения или составления задания.

В процессе выполнения задания в каждом эксперименте формируются файлы результатов и служебные файлы. Файлы результатов включают файлы экспериментальных данных, файл журнала экспериментатора и протокол работы САЭ. Служебными являются файлы описания начального и текущего состояния эксперимента и сеанса.

Таким образом, САЭ в отличие от ПОМС, в рамках которой она создается, включает лишь часть функционального наполнения ПОМС, а также файлы спецификации САЭ, задания на сеанс работы, служебные файлы и файлы результатов.

Рассмотрим более подробно компоненты резидентной части САЭ.

### 3. РЕЗИДЕНТНАЯ ЧАСТЬ СИСТЕМЫ

3.1. Основная управляющая программа. На рис.1 представлена программа, включающая несколько однотипных вложенных циклов, а на рис.2 - информационная модель этой программы, содержащая одинаковые по составу записи для каждого из циклов. В каждом цикле содержатся инструкции проверки наличия работы для данного уровня управления, загрузки (LM) и выполнения (ST) программы подготовки задания для вложенного уровня, выполнения вложенного цикла, а также загрузки и выполнения программы обработки результатов, полученных во вложенном цикле. Названия программ подготовки задания и обработки результатов содержатся в ИМ данного уровня /рис.2/. Управляющая программа, представленная на рис.1, является основной управляющей программой САЭ. Программы циклов в сочетании с виртуальными программами подготовки задания и обработки результатов, названия которых заданы в ИМ, являются процессорами соответствующих уровней управления /процессорами сеанса, эксперимента и т.д./.

3.2. Подсистема динамического распределения памяти. Инструкции загрузки (LM) и выполнения (ST) являются обращениями к подсистеме динамического распределения памяти/7/. Эта подсистема использует "кольцевой" буферный участок оперативной памяти раз-

```

VAR
  SESS:^CSDE;
  SAPR:NAME;
  ...
  BEGIN
    ...
    WITH SESS^ DO
      BEGIN
        WHILE SE.C <= SE.B DO
          BEGIN
            LM(SE.SPN);ST;
            WHILE EX.C <= EX.B DO
              BEGIN
                LM(EX.SPN);ST;
                WHILE PA.C <= PA.B DO
                  BEGIN
                    LM(PA.SPN);ST;
                    LM(SAPR);ST;
                    PA.C:=PA.C+1;
                    LM(PA.EPN);ST;
                  END;
                  PA.C:=PA.S;
                  EX.C:=EX.C+1;
                  LM(EX.EPN);ST;
                END;
                EX.C:=EX.S;
                SE.C:=SE.C+1;
                LM(SE.EPN);ST;
              END;
            END;
          END;
        END;
      END;
    END.
  
```

Рис. 1. Управляющая программа.

мером ~10К слов. При загрузке очередного потребовавшегося модуля память для него в буфере высвобождается по дисциплине FIFO. Перед загрузкой просматривается список присутствующих в буфере модулей, что устраняет избыточные операции загрузки. Инструкции ST инициирует выполнение модуля, найденного или загруженного по инструкции LM.

3.3. Информационная модель сеанса. ИМ сеанса включает ИМ процессоров, управляющих выполнением процессов во время сеанса, и состоит из фиксированной и изменяемой частей. Фиксированная часть содержит компоненты /записи/, структура которых не зависит от методики эксперимента. Примеры таких компонентов даны на рис.2. Изменяемая часть ИМ сеанса включает модели виртуальных процессоров. В ИМ виртуальных процессоров выделены массивы экспериментальных данных. Запоминание ИМ сеанса и восстановление

```

PROGRAM SESSION;
TYPE
  NAME = ARRAY [1..6] OF CHAR;
  DATI = RECORD
    D:INTEGER;
    T:REAL;
  END;
  STATE = RECORD
    N:NAME;
    S,C,B,E:INTEGER;
    SPN,EPN:NAME;
    COUN:REAL;
    LB,LE:DATI;
    STAT:INTEGER;
  END;
  CSDE = RECORD
    SE,EX,PA:STATE;
  END;
  
```

Рис. 2. Структура информационной модели управляющей программы.

ее образа в оперативной памяти при рестарте выполняется системными программами. Запоминание массивов экспериментальных данных и восстановление этой части ИМ выполняется функциональными программами.

ИМ является областью коммуникации всех программных модулей САЭ.

3.4. Резидентная библиотека подпрограмм /РБП/ введена для экономии расхода памяти /внешней и оперативной/, выделяемой независимо транслируемым программным модулем. Для использования языков Паскаль, Фортран-4, Макро-11 разработан способ и созданы средства, позволяющие:

- построить совокупную библиотеку системных /исполняющей системы ПАСКАЛЬ/ и пользовательских подпрограмм, составом которой управляет разработчик САЭ;
- обеспечить сохранение фиксированных адресов подпрограмм РБП при изменении остальных резидентных частей САЭ /в том числе и текста управляющей программы на языке Паскаль/;
- обеспечить доступ к этим подпрограммам из независимо транслируемых /включая и независимое редактирование связей/ программных модулей;
- предоставить возможность разработчикам САЭ и программных модулей принимать решение о помещении редко используемых библиотечных подпрограмм /например, из библиотеки системы ПАСКАЛЬ/ в тело программного модуля, а не в РБП.

Эти возможности реализованы без изменения используемой системы программирования на языке Паскаль путем разработки соответствующих служебных программ, используемых при построении резидентной части САЭ.

## ЗАКЛЮЧЕНИЕ

Основным результатом данной работы является комплекс средств создания САЭ, реализующий методику динамического формирования конфигурации программного обеспечения САЭ, обеспечивающий мобильность и комплексированность/8/ функциональных компонентов САЭ при изменении методики эксперимента и унификацию системного наполнения САЭ.

Отличительными особенностями САЭ, реализуемых в рамках данного комплекса, являются следующие:

1. В соответствии с предварительно составленным пользователями списком заданий система может автоматически выполнять сеанс, включающий последовательность различных экспериментов на различном составе оборудования, подключенного к данной ЭВМ.

2. Созданы системные программы обеспечения рестарта сеанса при сбоях и авариях.

3. Системными программами реализуется накопление данных для журнала экспериментатора.

Отдельные компоненты комплекса и разработанные методы обладают автономностью и могут быть использованы отдельно для решения частных проблем. Это относится, например, к программам ДРП, методике модульной организации программ на языке Паскаль, средствам управления составом резидентной библиотеки и др.

## ЛИТЕРАТУРА

1. Балука Г., Саламатин И.М. ОИЯИ, P10-85-200, Дубна, 1985.
2. Вагов В.А. и др. Труды XI Международного симпозиума по ядерн. электронике. ОИЯИ, Д13-84-53, Дубна, 1984, с.304.
3. Белушкин А.В. Автореферат диссертации. ОИЯИ, 14-84-347, Дубна, 1984.
4. Ершов А.П., Ильин В.П. В кн.: Пакеты прикладных программ: проблемы и перспективы. "Наука", М., 1982, с.4.
5. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12546, Дубна, 1979.
6. Балука Г., Саламатин И.М. ОИЯИ, 10-84-573, Дубна, 1984.
7. Балука Г. ОИЯИ, P10-84-789, Дубна, 1984.
8. McCall J.A., Matsumoto M. Final Tech.Rep., June 1978-July 1979, 1980, vol.1, VII.

Рукопись поступила в издательский отдел  
18 апреля 1985 года.

Балука Г., Саламатин И.М.

P10-85-281

Динамически формируемая программная система автоматизации экспериментов

Разработан комплекс средств, реализующий методику динамического формирования конфигурации программного обеспечения систем автоматизации экспериментов /САЭ/. Обеспечена комплексированность функциональных компонентов при изменении методики эксперимента и унификация системного наполнения САЭ. В соответствии с предварительно составленным пользователем списком заданий САЭ может автоматически выполнять сеанс, включающий последовательность различных экспериментов на различном составе оборудования, подключенного к данной ЭВМ. Созданы системные программы обеспечения рестарта сеанса при сбоях и авариях. Системными программами реализуется накопление данных для журнала экспериментатора. Отдельные компоненты комплекса и разработанные методы обладают автономностью и могут быть использованы отдельно. Это относится к программам динамического распределения памяти, методике модульной организации программ на языке Паскаль, средствам управления составом резидентной библиотеки и др. Комплекс работает под управлением ОС RT-11.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1985

Перевод О.С.Виноградовой

Baluka G., Salamatin I.M.

P10-85-281

A Dynamically Formed Software System for Data Acquisition

The set of utilities that realizes the algorithm of dynamical creating of program configuration for data acquisition systems has been developed. The set allows one to build program system which does not need changes to follow variations in experiment methodology utilizing the set of unified programs. The measuring session (which can consist of sequence of measurements for different experimenters working with different hardware) is performed in accordance with the list of tasks created by the user before the measurements. The system programs can perform resuming of measurements after system's crash. The system creates the journal file of its work. Different subsystems of the set can be used apart. It can be applied to the memory control subsystem, the organization of the utility procedures, the utilities for the resident library control. The set works under control of the RT-11 operating system.

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

Preprint of the Joint Institute for Nuclear Research. Dubna 1985