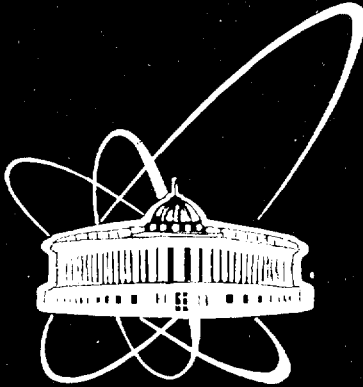




XJ9900286



**ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ**

Дубна

P10-99-65

Е.А.Горская, В.Н.Самойлов

**МЕТОД ПОСТРОЕНИЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ МНОГОКАНАЛЬНОЙ СИСТЕМЫ
АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ
ФИЗИЧЕСКИМИ ЭКСПЕРИМЕНТАМИ
НА БАЗЕ ИНСТРУМЕНТАЛЬНОГО ПАКЕТА
NATIONAL INSTRUMENTS LABVIEW**

Направлено в журнал «Автоматизация проектирования»

ℓ

30 - 45

1999

1. Введение

Пакет LabVIEW (Laboratory Virtual Instrument Engineering Workbench) американской фирмы National Instruments представляет в качестве средства создания приложений не традиционный текстовый язык программирования, а графический язык, который, обладая всеми возможностями, скажем, языка С, имеет мощные средства для быстрого создания графического интерфейса разрабатываемого приложения, а именно, панелей управления и индикации для физических и технологических процессов, с одной стороны, и быстрого составления программ ввода и обработки данных в виде блок-диаграмм, с другой стороны. Все это компилируется в эффективный машинный код.

Несомненно, LabVIEW является идеальной средой для создания систем автоматизированного управления установками при проведении физических экспериментов. Большой набор графических индикаторов и объектов управления, широкий спектр библиотечных программ, возможность применения принципов структурирования и модульности при программировании, наконец сам процесс построения блок-диаграмм дают возможность сделать трудоемкий процесс программирования занимательным и быстрым [2-8].

Однако при разработке более или менее сложной системы, в которой число сигналов управления превышает несколько десятков, возникает необходимость в создании общего подхода для программирования отдельных объектов управления. В данной работе изложен один из возможных подходов, который был разработан и применен в процессе создания автоматизированной системы управления ECR-источником ионов.

2. Гипотетическая модель

Поскольку LabVIEW является проблемно-ориентированной средой программирования, то, естественно, и подход к построению модели выбираем объектно-ориентированный.

Постановка задачи

Если проанализировать объект моделирования - автоматизированную систему управления как физическими процессами так, впрочем, и технологическими, то можно выделить два класса абстракций [1].

- Множество физических объектов, которые выполняют функции управления, измерения и контроля при проведении физического эксперимента. К ним относятся средства связи с ЭВМ, аналоговое и

цифровое оборудование, измерительные системы и т. д., которые обозначим как множество реальных объектов {RealObj}.

- Множество графических двойников этих физических объектов, созданных на панелях управления (Front Panel), на экране компьютера, через которые оператор и управляет процессами. Эти графические объекты обозначим как множество виртуальных объектов {VirtObj}.

С учетом требований к удобствам в управлении физическими процессами соответствие между реальными и виртуальными объектами может и не быть взаимно однозначным. Любой физический (реальный) объект может быть изображен (представлен виртуально) на нескольких панелях управления.

Задача состоит в том, чтобы

- При выполнении управляющих воздействий на графические объекты из множества {VirtObj} эти воздействия выполнены бы на физических объектах управления из множества {RealObj}. Время передачи управляющих воздействий не должно превышать определенного (приемлемого) промежутка времени. Обозначим этот промежуток как время передачи управляющих воздействий - $dTcnt$. Таким образом, операция {VirtObj}→ {RealObj} должна выполняться за время $dTcnt$.
- Состояние объектов из множества {RealObj} отображалось бы на состояние объектов из множества {VirtObj} с определенной (приемлемой) периодичностью. Обозначим промежуток времени между двумя обновлениями состояний объектов как период обновления индикации - $dTind$. Таким образом, операция {VirtObj}← {RealObj} должна выполняться за время $dTind$.

Очевидно, что время реакции системы: $Treak=dTcnt+dTind$, и желательно, чтобы это время было минимальным и соразмеримым со временем выполнения процессов в физическом оборудовании.

2.1. Описание модели.

Для того, чтобы выполнить поставленную задачу, очевидно, надо выделить (создать) еще один класс абстракций, который бы и выполнял задачу коммуникации между реальными и виртуальными объектами.

Добавим к множествам объектов {VirtObj} и {RealObj} еще одно множество объектов, которые и будут служить средством коммуникации между объектами множеств {VirtObj} и {RealObj}. Назовем это множество множеством информационных каналов и обозначим его как множество {InfCannel}.

Определим над множествами {InfCannel}, {VirtObj} и {RealObj} набор методов в виде следующих множеств [1] :

- множество функций {setVirtCann(VirtObj, InfCannel)}, обеспечивающих передачу информации от виртуального объекта управления в информационный канал;

- множество функций {getVirtCann(VirtObj, InfCannel)}, обеспечивающих передачу информации из информационного канала виртуальным объектам индикации и измерения;
- множество функций {outRealCann(RealObj, InfCannel)}, обеспечивающих вывод информации (сигналов управления) из информационных каналов во внешние устройства управления;
- множество функций {inRealCann(RealObj, InfCannel)}, обеспечивающих ввод информации с внешних устройств контроля и измерений в информационные каналы.

2.2. Свойства объектов и отношение между объектами

Исходя из реальных задач системы управления, можно определить некоторые свойства объектов и правила их поведения, что сделает модель более простой, конкретной и корректной.

Условия, которым должны подчиняться объекты:

- каждый объект из {VirtObj} и {RealObj} может быть только или источником или приемником;
- каждый объект из {InfCannel} всегда обеспечивает движение информации только в одну сторону;
- между информационными каналами {InfCannel} и реальными объектами {RealObj} должно быть взаимно однозначное соответствие, причем каждый физический объект жестко связан с одним информационным каналом;
- виртуальные объекты {VirtObj} связаны с информационными каналами свободно, по принципу "многие к одному";
- если несколько виртуальных объектов управления связаны с одним и тем же информационным каналом, то при передаче информации по каналу необходимо выполнять механизм взаимной блокировки доступа к каналу, используя систему приоритетов виртуальных объектов;
- виртуальные объекты, если это необходимо, должны иметь возможность удаленного доступа к информационным каналам, т.е. функции из множеств {outRealCann(RealObj, infCannel)} и {inRealCann(RealObj, infCannel)} должны поддерживать сетевые протоколы [7];
- множества {VirtObj}, {RealObj}, {InfCannel} должны быть открытыми, с тем чтобы при изменении стандартного аппаратно-технического оборудования, его наращивания и усложнения, систему можно было бы быстро настроить, дополняя множества {VirtObj}, {RealObj} и {InfCannel} без изменения программного кода, и только при включении в управление аппаратных средств нового стандарта возникла бы необходимость в разработке новых программных модулей – функций обмена с новыми устройствами и включения их во множества {outRealCannel()} и (или) {inRealCannel()};

- дисциплина доступа к информационным каналам должна обеспечивать однозначность передаваемой информации с помощью синхронизации операций обмена между объектом-источником и объектом-приемником.

2.3. Эффективность предлагаемой модели

Построение системы управления по предлагаемой модели дает следующие качественные достоинства:

- можно построить систему с практически неограниченным числом управляющих сигналов;
- возможность быстрого изменения системы управления и отображения при изменении конфигурации физической аппаратуры;
- возможность перенастройки системы управления в ходе проведения эксперимента;
- независимость информационных каналов друг от друга;
- возможность параллельного выполнения операций обмена с внешними устройствами, обработки и отображения информации. Тем самым можно получить время реакции системы, требуемое для проведения эксперимента;
- возможность построения сетевого варианта автоматизированной системы управления, причем в этом случае графические объекты из множества {VirObj} выступают в роли клиентов и располагаются на различных машинах (рабочих станциях), а объекты из множеств {RealObj} и {InfCannel} располагаются на машине-сервере [7] .

3. Архитектурные решения

3.1. Движение информационных потоков

Схема передачи и обработки информации в проектируемой модели представлена на рисунке.

Интерфейс с пользователем-оператором осуществляется через графические панели управления в понятиях LabVIEW.

Число панелей зависит от сложности установки. Целесообразно каждую отдельную подсистему выводить на свою отдельную панель, а общую индикацию и общее управление возложить на главную панель.

Таким образом, оператор может настраивать отдельные подсистемы, открывая панели управления, соответствующие этим подсистемам, а общее управление и общее наблюдение выполнять с главной панели.

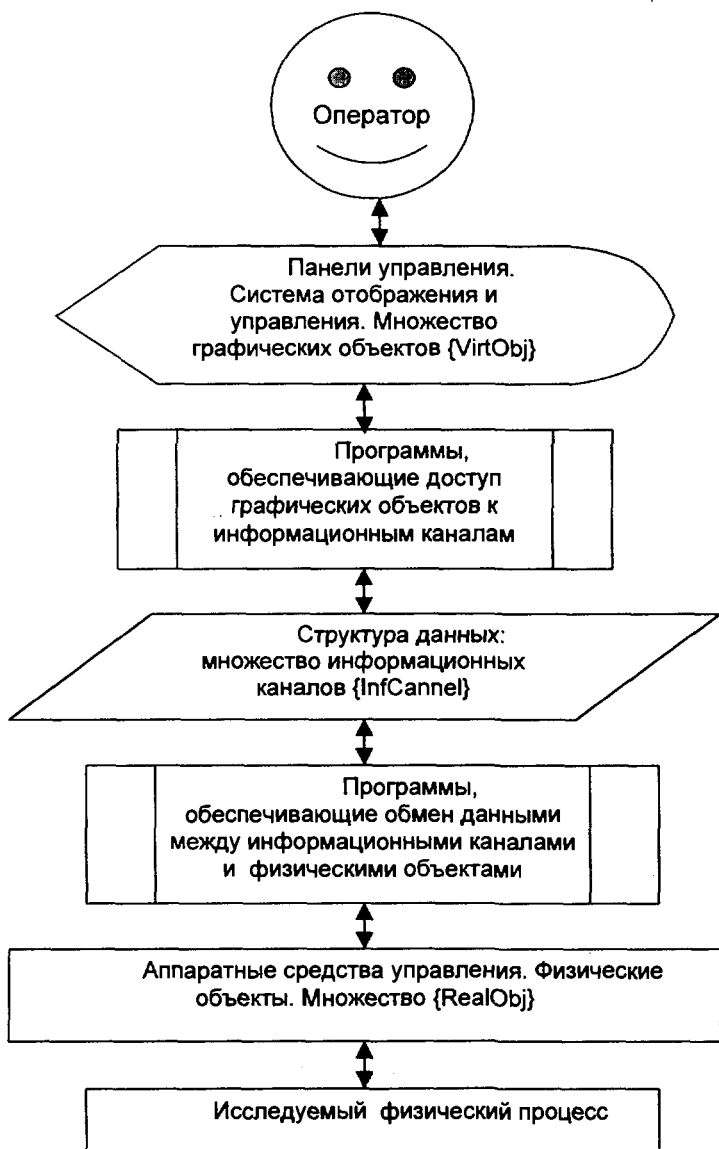
Если в эксперименте участвуют несколько разных специалистов, то можно каждому предоставить отдельную панель с интересующими его объектами.

С точки зрения программирования каждая панель является самостоятельной задачей - vi-программой. Распараллеливание их выполнения обеспечивается программной средой LabVIEW. vi-

программы, обслуживающие панели управления, связывают виртуальные объекты с информационными каналами с помощью функций из множеств {setVirtCann(VirtObj,InfCann)} и {getVirtCann(VirtObj,InfCann)}.

Множество информационных каналов описывается как глобальная структура данных – в виде таблицы, которая доступна для всех Vi-программ. Перечень и описание основных полей информационного канала приводятся ниже.

Программы, обеспечивающие обмен данными между информационными каналами и физическими объектами, наоборот представляют собой единый вычислительный процесс. Обслуживание идет последовательно, канал за каналом. Программы выполняют операции опроса состояния аппаратуры и передачи в тракт управляющих сигналов. Для этого программы используют функции из множеств {outRealCann(RealObj,InfCann)}, {inRealCann(RealObj,InfCann)}.



Среда LabVIEW

Рис. Схема обмена информацией между оператором и экспериментальной установкой

3.2. Структура информационных каналов

Множество информационных каналов {InfCannel} описывается как структура данных в виде таблицы. Каждому каналу соответствует запись в таблице. Набор данных-полей в записи включает в себя свойства канала, его текущее состояние и передаваемые данные.

Поля, описывающие свойства канала, являются статическими. Свойства канала не изменяются в процессе работы системы и задаются изначально при включении канала в систему. К этой группе свойств следует отнести следующие поля:

- Name – поле, содержащее словесное описание (или название) физического устройства - объекта из множества {RealObj}, который связан с данным каналом;
- IdentName - идентификатор канала, служит глобальной ссылкой как на канал, так и на физическое устройство, связанное с данным каналом;
- Type – тип канала – канал может быть аналоговым (“Analog”) или цифровым (“Logical”) в соответствии с типом внешнего устройства;
- Way - направление движения информации в канале. Значение “In” указывает на движение данных в сторону отображения:

{VirtObj} ← {InfCannel} ← {RealObj}.

Значение “Out” указывает на движение данных в сторону физического оборудования:

{VirtObj} → {InfCannel} → {RealObj};

- Address – это поле (или группа полей) определяет точный физический адрес устройства, на котором выполняется операция обмена. Например, если подключение выполнено через крейт КАМАК [9], то должны быть определены такие поля, как C - номер крейта КАМАК, N – номер станции в крейте КАМАК, A - адрес подстанции в блоке КАМАК, W – номер логической ножки, IstDat - номер цифрового регистра на устройствах связи с физическими объектами и т.п.
- Transform – поле (или группа полей), определяющее функции преобразования сигнала при обмене ЭВМ с физической аппаратурой. В эту группу могут включаться такие параметры, как Winv - признак инверсного или прямого кода при обмене данными, Maska - маска ввода или вывода данных, Coeff – значение калибровочного коэффициента усиления и т. д.
- Method – поле (или группа полей), определяющее метод (методы) доступа к каналу. Здесь может быть указана одна или несколько функций из множеств {setVirtCann(VirtObj,InfCannel)}, {getVirtCann(VirtObj,InfCannel)}, {setRealCann(VirtObj,InfCannel)} и {getRealCann(VirtObj,InfCannel)}.
- T/out – поле (или группа полей), определяющее всякого рода задержки при выполнении операции обмена на устройстве.

Содержимое полей, в которые записываются параметры текущего состояния канала, изменяется в процессе функционирования системы в соответствии с происходящими событиями. Исходное состояние канала задается также при включении канала в систему. В эту группу следует отнести следующие поля:

- StatusCannel – состояние канала – поле, через которое происходит синхронизация доступа к каналу. Может принимать значения "On" или "Off". Используется для синхронизации обмена данными через канал: объект-источник записывает данные в канал и открывает его для чтения, объект-приемник читает данные, закрывает канал для чтения и открывает его для записи информации;
- StatusDev – состояние внешнего устройства, к которому принадлежит данный канал. Может принимать значения "A" – устройство в автономном состоянии, "Free" – устройство свободно, "Busy" – устройство занято;
- FrontPanel – поле, в которое записывается идентификатор панели управления, графический объект которой связан в данный момент с каналом.

Поля для передачи данных могут использоваться как в качестве регистров, так и в качестве ссылок на буферы с данными. Объект-источник записывает данные на регистр или в буфер, а объект-приемник эти данные читает.

3.3. Описание вычислительного процесса

При открытии любой из управляющих панелей происходит привязка графических объектов, изображенных на панели к информационным каналам, а стало быть, и к своим физическим двойникам.

Вычислительный процесс организуется двумя независимыми циклами: циклом обмена данными между графическими объектами и информационными каналами и циклом обмена данными между информационными каналами и физическими объектами управления.

В цикле обмена данными между графическими объектами и информационными каналами процедуры обмена данными на разных каналах выполняются параллельно, независимо друг от друга.

В цикле обмена данными между физическими объектами и информационными каналами процедуры обмена с внешними устройствами могут выполняться и параллельно, и последовательно, в зависимости от типа подключения устройств.

4. Заключение

На основе вышеизложенной модели реализован комплекс программ для автоматизированной системы управления ECR–источником ионов. Аппаратно управление реализовано на базе двух крейтов КАМАК, последовательно соединенных через контроллер КК011 и интерфейсную плату ПИ021. Для управления, индикации и измерения использовано 120 цифровых и 50 аналоговых сигналов.

Управление ECR–источником осуществляется с семи управляющих панелей. Время реакции системы – 1-1,5 секунды. Время перехода с одной панели на другую –1 секунда.

Авторы выражают благодарность за сотрудничество и содействие в работе В.Б. Кутнеру., В.Н. Логинову, В. О. Громову.

Литература

1. Г.Буч, Объектно-ориентированное программирование с примерами применения, перевод с английского, 1992.
2. LabVIEW, Tutorial for Windows, National Instruments Corporation, September 1994.
3. LabVIEW, User Manual, National Instruments Corporation, September 1994.
4. LabVIEW, Function Reference Manual, National Instruments Corporation, September 1994.
5. LabVIEW, Code Interface Reference Manual, National Instruments Corporation, September 1994.
6. LabVIEW, Data Acquisition VI, Reference Manual, National Instruments Corporation, September 1994.
7. LabVIEW, Networking, Reference Manual, National Instruments Corporation, September 1994.
8. LabVIEW, Utility VI, Reference Manual, National Instruments Corporation, September 1994.
9. Система КАМАК ГОСТ 26.201.80. Изд. стандартов, М.,1980.

Рукопись поступила в издательский отдел
17марта 1999 года.

Горская Е.А., Самойлов В.Н.

P10-99-65

Метод построения программного обеспечения
многоканальной системы автоматизированного управления
физическими экспериментами
на базе инструментального пакета National Instruments LabVIEW

В настоящей работе описывается метод построения автоматизированной системы управления технологическими процессами в интегрированной программной среде LabVIEW. На основе объектно-ориентированного подхода в проектировании сложных систем строится гипотетическая модель метода построения программного обеспечения автоматизированной системы управления физическими экспериментами. На базе этой модели описаны архитектурные решения и показаны пути реализации предложенного метода.

Работа выполнена в Лаборатории вычислительной техники и автоматизации и Научном центре прикладных исследований ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна, 1999

Перевод авторов

Gorskaya E.A., Samoilov V.N.

P10-99-65

The Software Developing Method
for Multichannel Computer-Aided System
for Physical Experiments Control, Realized by Resources
of National Instruments LabVIEW Instrumental Package

This work is describing the method of developing the computer-aided control system in integrated environment of LabVIEW. Using the object-oriented design of complex systems, the hypothetical model for methods of developing the software for computer-aided system for physical experiments control was constructed. Within the framework of that model were described architecture solutions and implementations of suggested method.

The investigation has been performed at the Laboratory of Computing Techniques and Automation and at the Scientific Center for Applied Researches, JINR.

Preprint of the Joint Institute for Nuclear Research, Dubna, 1999

Редактор Е.Ю.Шаталова, Макет Н.А.Киселевой

Подписано в печать 22.04.99
Формат 60 × 90/16. Офсетная печать. Уч.-изд. листов 0,48
Тираж 345. Заказ 51334. Цена 58 к.

Издательский отдел Объединенного института ядерных исследований
Дубна Московской области