

нал выводится диагностическое сообщение. Возможна декларация запроса L по предварительно сформированному подпрограммой CDREG КАМАК-адресу и настройка пакета, как описано в п. 7.

Для многопрограммных операционных систем старших моделей ЭВМ типично создание так называемых драйверов — модулей операционной системы, монополизирующих взаимодействие с определенным подмножеством оборудования и управляемых программными запросами. Для микро-ЭВМ с ограниченной оперативной памятью такой подход неприемлем, так как драйвер, обладающий полным набором возможностей, занимает слишком много места в ОЗУ, а подготовка и настройка специализированного драйвера с ограниченными возможностями для каждой конкретной конфигурации установки слишком трудоемки и требуют высокой квалификации. Кроме того, каждое обращение к драйверу связано с обработкой запроса операционной системой и занимает 4—5 мс.

Альтернативой драйверу является следующий избранный нами метод: с аппаратурой взаимодействуют непосредственно подпрограммы прикладных программ. В тех случаях, когда вмешательство операционной системы или передача управления другим программам нежелательны, работа программа увеличивает приоритет на время, допускаемое операционной системой (100—500 мкс), запрещая тем самым прерывания ее работы. В частности, одиночная КАМАК-команда выполняется в три этапа: вначале запись кода команды в регистр управления крейт-контроллера (CSR), затем организация КАМАК-цикла в соответствии с кодом команды, записанным в CSR, и, наконец, считывание CSR в ЭВМ для анализа ответов X и Q. Все три этапа должны составлять непрерываемую последовательность, иначе выполнение операции окажется неправильным в результате вмешательства других программ. При блочных передачах несколько таких циклов выполняются последовательно, занимая процессор на допускаемое системой время. Увеличение этого времени привело бы к ухудшению времени реакции на события в аппаратуре, а уменьшение — к падению эффективности.

#### ЛИТЕРАТУРА

1. SUBROUTINES FOR CAMAC.— ESONE/SR/01, Sep. 1978.
2. Казакова Н. А., Панкрац Е. В. Реализация языка промежуточного уровня IML на ЭВМ типа СМ-3.— Автометрия, 1980, № 3.
3. The Definition of IML, a Language for Use in CAMAC System.— ESONE/IML/01, ESONE SECRETARIAT, Oct. 1974.
4. Sendall D. M. Minicomputer Software for High-Energy Physics Experiments.— Proc. of the 1978 CERN School of Computing, June 1978.

*Поступила в редакцию 11 декабря 1981 г.*

УДК 681.3.06

**М. А. КУРИЛОВ, В. В. МАНАКО, А. И. НИКИТИН,  
И. В. ЧИЧКАНЬ**

*(Киев)*

### **СТАНДАРТНЫЙ ГРАФИЧЕСКИЙ ПАКЕТ ГРАС. СРЕДСТВА ОТОБРАЖЕНИЯ, ХРАНЕНИЯ И ПЕРЕДАЧИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ**

С появлением ВЦКП и сетей ЭВМ на первый план в области создания и развития графического программного обеспечения выдвинулась проблема стандартизации. Она подразумевает разработку стандартного графического пакета, стандартных форм представления и способов пе-

передачи графической информации в сети, нахождение оптимального способа распределения нагрузки между локальным графическим процессором и обслуживающей ЭВМ, синхронизацию параллельных процессов и др.

Эта задача становится безотлагательной в связи с непрерывным и интенсивным ростом удельного веса машинной графики в системах автоматизации проектирования и обработки экспериментальных данных, а также в связи с тем, что до настоящего времени графические системы оказываются, как правило, не совместимы друг с другом [1].

В Институте кибернетики АН УССР разработан стандартный графический пакет ГРАС (графический стандарт) [2—4], удовлетворяющий международным рекомендациям по стандартизации ACM/SIGGRAPH [5, 6].

В статье рассмотрены структура и функциональные возможности ГРАС по отображению, хранению и передаче графической информации.

**Проектирование ГРАС.** Основной целью проектирования ГРАС является содействие совместному использованию графических программ, т. е. переносимости этих программ из одной вычислительной системы в другую. Однако абсолютная переносимость графических программ, по-видимому, не достижима в ближайшем будущем [6, 7]. Ключевым признаком переносимости прикладных программ является то, что их транспортировка не требует понимания структуры программы.

Второй целью проектирования является содействие адаптируемости программиста, т. е. максимальному облегчению для него перехода от одной графической системы, надстраиваемой над ГРАС, к другой, что в отсутствие стандартизации требует изучения новых соглашений, техники, понятий и т. п.

ГРАС имеет следующую двухуровневую структуру:

1. *Базисный уровень* обеспечивает переносимость прикладных графических программ. С учетом применения ГРАС в широком диапазоне графических приложений и устройств отображения специфицируются два класса подуровней базисного уровня: для вывода и ввода. К классу вывода относятся следующие три подуровня:

базисный вывод, поддерживающий двух(2D)- и трехмерные (3D) примитивы вывода и их атрибуты; временные сегменты, т. е. сегменты, не имеющие записей, содержащих примитивы вывода, и их атрибуты; преобразования проектирования (ПРП); управление графическими устройствами;

буферизованный вывод, представляющий собой базисный вывод, дополненный возможностями сохранения сегментов и их атрибутов;

динамический вывод, поддерживающий полный набор возможностей описания изображений, включая преобразования изображений, такие как масштабирование, поворот и сдвиг.

К классу ввода относятся следующие два подуровня.

Нет ввода. Этот подуровень используется для приложений, требующих только вывод.

Есть базисный ввод. Обеспечивает интерактивные возможности графических устройств, поддерживает примитивы ввода.

2. *Универсальные предметные графические элементы и функции* (УПГЭФ) описывают типовые, цельные графические элементы и функции, их свойства, допустимое манипулирование ими и их конструирование. Введение в ГРАС этого уровня объясняется следующим. Во-первых, в [5] обосновано, что в базисный уровень целесообразно включать функции, которые относятся строго к генерации изображений, а функции по конструированию графических объектов и манипулированию ими передавать более высоким уровням (данный подход успешно развивается в [5—7]). Во-вторых, в настоящее время не определены примитивы более высокого уровня, например вывода карт, кривых, знаков, ввода и др. [6, 7]. Однако оказалось возможным выделить типовые, цельные графические конструкции, которые и образуют уровень УПГЭФ.

Отметим, что ГРАС ориентирован в основном на использование графических регистрирующих устройств и векторных регенерационных дисплеев и удовлетворяет принципам и критериям, сформулированным в [2—4].

**Функциональные возможности базисного вывода и УПГЭФ ГРАС.** Функциональные возможности базисного вывода ГРАС могут быть кратко описаны при помощи следующих категорий.

*Примитивы вывода.* Графические объекты описываются в объектных координатах при помощи вызовов подпрограмм порождения 2D-или 3D-примитивов вывода. Ими являются перевод пишущего элемента (пера или луча), рисование отрезка прямой линии, ломаной, маркера, последовательности маркеров и строки знаков текста. Текущая позиция (ТП) пишущего элемента используется как начальная точка для линий и строк текста. Параметры подпрограмм порождения примитивов вывода могут специфицироваться как в абсолютных, так и в относительных координатах (относительно ТП).

*Атрибуты примитивов вывода.* Атрибуты, точнее значения атрибутов, воздействуют на проявление примитивов вывода. После создания примитива вывода его атрибуты можно изменить только путем удаления сегмента и повторной спецификации примитива. Поддерживаются следующие атрибуты примитивов: цвет, яркость, тип линии (сплошная, пунктирная и т. д.) и ее толщина, номер маркера, номер шрифта знаков, размеры знака, качество строки текста, расстановка знаков строки и ориентация строки текста. Атрибут качества строки текста показывает степень воздействия остальных атрибутов текста на примитив. Для низкого качества текста всегда может использоваться аппаратный генератор символов, для среднего качества — лишь в некоторых случаях, а для высококачественного — практически не применяется.

*Сегментация картин.* Все примитивы вывода должны помещаться в сегмент, специфицированный в прикладной программе. Сегмент описывает образ, который является изображением некоторого объекта и частью картины, воспроизводимой на зрительной поверхности, например на экране дисплея. Прикладная программа описывает объект путем создания (открытия) сегмента, вызовов подпрограмм порождения примитивов вывода и их атрибутов и затем закрывает сегмент.

*ПРП.* Для 2D-объектов ПРП специфицируется при помощи окна и поля индикации [2, 3] (оба — прямоугольники) на выбранной зрительной поверхности. Окно может быть повернуто относительно главных осей объектной системы координат; оно используется при выполнении операций отсечения и отображения части объекта в поле индикации. Для 3D-объектов ПРП специфицируется при помощи точки зрения, типа проекции (параллельная или перспективная), окна в проекционной плоскости и поля индикации, которое может быть и прямоугольным параллелепипедом. ПРП допускает произвольное позиционирование точки зрения и окна. Объекты могут отсекаются границами проекционного объема — прямоугольной пирамидой для перспективной проекции или параллелепипедом для параллельной проекции, а затем проецируются на проекционную плоскость и далее отображаются в поле индикации. Конечный подобъем проекционного объема специфицируется при помощи двух плоскостей — передней и задней, параллельных проекционной плоскости. Отсечения окном и этими плоскостями управляются отдельно. ПРП являются достаточно общими, так как центральные линии проекционного объема не обязательно перпендикулярны проекционной плоскости. Таким образом, ортогональная, кабинетная и другие проекции представляют собой специальные случаи параллельной проекции.

*Управление.* Подпрограммы управления обеспечивают инициализацию и завершение работы ГРАС, выбор одной или более логических зрительных поверхностей для вывода, устанавливают механизм обработки ошибок. Подпрограмма «Новый кадр» дает возможность одновременно удалять все временные сегменты с выбранных зрительных поверхно-

Для обеспечения переносимости прикладных программ на функциональном уровне имеется «защита» от специфических характеристик графических устройств. Пакет отвечает развитому принципу «умолчания», т. е. если прикладная программа не специфицирует некоторых параметров или группы параметров, то автоматически устанавливаются их наиболее вероятные значения.

Базисный вывод ГРАС содержит около 95 подпрограмм функциональной спецификации.

Практически во всех руководствах по графическим пакетам отмечается, что они рассчитаны на опытных программистов. Несмотря на качественное отличие ГРАС от ранее разработанных пакетов, использование его базисного уровня программистами, не владеющими навыками графического программирования, может быть связано с определенными трудностями.

Рассмотрим в качестве примера одну широко распространенную задачу: построение графика таблично заданной функции. В общем случае программист должен выполнить следующие действия: 1) инициализировать графический пакет и 2) графическое устройство; 3) выполнить масштабирование исходных данных; 4) построить оси декартовой (или другой, например, логарифмической) системы координат с оцифровкой, а также 5) кривую, аппроксимирующую исходную функцию, и 6) различные пояснительные надписи; 7) завершить использование графического устройства и 8) графического пакета.

Общие соображения по реализации действий 1—8 следующие. Каждое действие обычно реализуется одной или несколькими подпрограммами графического пакета, причем некоторые действия требуют проведения предварительных вычислений. Хорошо известно также, что совместное применение графических подпрограмм предполагает наличие дополнительных требований по индивидуальному их использованию. Эти трудности реализации действий 1—8 можно конкретизировать следующим образом. Действия 1, 2, 7, 8 необходимо выполнять в любой прикладной графической программе. Таким образом, первая трудность возникает уже при стыковке вычислительной и графической частей прикладной программы. Вторая трудность — отсутствие в общем случае стандартизации на уровне программирования, т. е. отсутствуют примитивы высокого уровня, реализующие, например, пп. 4, 5.

Поэтому для широкого круга программистов предлагаются средства (уровень УПГЭФ) и метод, обеспечивающие простоту как стыковки вычислительной и графической частей программы, так и написания графической части программы, т. е. легкость и естественность общения с ЭВМ. В некоторых случаях предлагаемый метод предоставляет возможность более быстрой отладки всей программы, выбора варианта решения проблемной задачи и т. д.

На уровне УПГЭФ программисту предоставляется обширный набор графических конструкций и преобразований, таких как «График функции», «Гистограмма», «Изображение пересечения двух и более поверхностей» и т. д. Обращение к подпрограммам УПГЭФ, например, на ФОРТРАНе ОС ЕС выполняется на уровне оператора CALL, как к обычным подпрограммам, при этом никаких дополнительных действий (и знаний) от программиста не требуется. Например, в составе УПГЭФ имеется подпрограмма GRAF12 для построения графика таблично заданной функции в декартовой системе координат. Формат вызова подпрограммы следующий:

CALL GRAF12(X, Y, N, ITX, IX, ITY, IY, ITZ, IZ, W, H),  
X, Y — одномерные массивы абсцисс и ординат N опорных точек; (ITX, IX), (ITY, IY), (ITZ, IZ) — три пары параметров, специфицирующих надписи соответственно над осями абсцисс, ординат и над графиком. Первый параметр каждой пары — массив символов надписи, а второй указывает число символов надписи, W, H — ширина и высота прямо- и гистограмм в различных системах координат с наглядной компоновкой и масштабированием;

построение изображений поверхностей (в том числе пересекающихся с плоскостями и другими поверхностями и удаление при этом невидимых линий) и пространственных кривых;

построение элементов машиностроительных чертежей, например рабочих чертежей отдельных конструкций колес гидравлических машин;

построение некоторых объектов (например, лопастей турбин), находящихся в аэродинамических условиях.

УПГЭФ ГРАС содержит около 50 подпрограмм функциональной спецификации.

Следует отметить, что выбор параметров УПГЭФ мы рассматривали как процесс дальнейшего развития категорий базисного вывода ГРАС с использованием критериев выбора функций, описанных в [2, 3].

**Хранение и передача графической информации в ГРАС.** С точки зрения реализации ГРАС в сети ЭВМ каждому из подуровней базисного уровня ГРАС соответствует графический протокол определенного уровня, который образуется расширением предыдущего. В основе этой иерархии графических протоколов находится протокол базисного вывода, обеспечивающий минимальные возможности по работе с графическими устройствами. Отметим, что команды графического протокола, например базисного вывода, практически совпадают с командами драйверов устройств. Графические протоколы входят в общую иерархию сетевых протоколов, поэтому их реализация базируется на протоколе «процесс — процесс», который обеспечивает связь в сети отдельных процессов, в частности прикладных графических программ.

Передача графической информации в сети ЭВМ осуществляется в виде стандартного устройство-независимого псевдодисплейного файла обмена (ПДФО). Обмен данными в формате ПДФО потребовал разработки двух программных компонент — генератора и транслятора ПДФО. ПДФО не является средством хранения и восстановления изображений или сохраняемых сегментов в рамках прикладной графической программы. Он может применяться как средство передачи изображений между двумя различными ЭВМ; употребляться для проверки формирования графических данных; позволять делать «твердую» копию изображений, получаемых в процессе диалога; служить средством хранения графической информации, т. е. основой для создания графических баз данных; быть средством для установления эквивалентности и верификации графических данных; использоваться как стандартный интерфейс для интеллектуальных терминалов.

Процесс генерации изображений укрупненно можно представить в следующем виде: прикладная программа — устройство-независимая графическая система — подпрограммы драйвера устройства — графическое устройство. Генерация ПДФО в такой системе выполняется следующим образом: прикладная программа — устройство-независимая графическая система — генератор ПДФО — ПДФО. Графический вывод возможен двумя путями: ПДФО — транслятор ПДФО — устройство-независимая графическая система — подпрограммы драйвера устройства —

графическое устройство; ПДФО — транслятор ПДФО — подпрограммы драйвера устройства — графическое устройство.

Практическая реализация ПДФО ГРАС осуществляется в рамках создания общего программного обеспечения сети вычислительных центров АН УССР.

**Заключение.** В настоящее время ГРАС является ядром графического пакета научно-исследовательского программно-технического комплекса для моделирования САПР ИК АН УССР и пакета подпрограмм для графического отображения результатов обработки научной информации в вычислительных центрах АН УССР.

#### ЛИТЕРАТУРА

1. Баяковский Ю. М., Галактионов В. А. Графические протоколы.— Автометрия, 1978, № 5.
2. Курилов М. А., Манако В. В., Никитин А. И. Некоторые вопросы стандартизации программного обеспечения графических систем.— Киев: изд. ИК АН УССР, 1981. (Препринт/АН УССР, Ин-т кибернетики; № 81—15).
3. Манако В. В. Стандартный графический пакет ГРАС. Проектирование. Базис вывода.— Управляющие системы и машины, 1981, № 3.
4. Курилов М. А., Манако В. В., Никитин А. И., Чичкань И. В. Стандартный графический пакет ГРАС. Средства отображения, хранения и передачи графической информации.— В кн.: Тез. докл. VI Всесоюз. конф. «Автоматизация научных исследований на основе применения ЭВМ (Новосибирск, 1981)». Новосибирск: изд. ИАиЭ СО АН СССР, 1981.
5. ACM/SIGGRAPH: Status Report of the Graphics Standards Planning Committee of ACM/SIGGRAPH.— Computer Graphics, 1977, vol. 11, N 3.
6. ACM/SIGGRAPH: Status Report of the Graphics Standards Planning Committee.— Computer Graphics, 1979, vol. 13, N 3.
7. Newman W. M., Sproul R. F. Principles of Interactive Computer Graphics/Sec. ed. McGraw-Hill.— N.-Y., 1979, p. 541.

*Поступила в редакцию 29 июня 1981 г.*

УДК 681.44

**В. Н. ВЕРХОТУРОВ, В. И. ЗАБОЛОТСКИХ**

*(Москва)*

### **АВТОМАТИЗИРОВАННЫЙ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ КОМПЛЕКС ЭПР-СПЕКТРОСКОПИИ**

**Введение.** Методы электронного парамагнитного резонанса (ЭПР) являются весьма информативными и широко распространенными методами исследования структуры различных веществ и тканей. Регистрируемое в эксперименте явление резонанса наблюдается при совпадении частоты переходов между уровнями электронных магнитных моментов молекул в постоянном магнитном поле с частотой приложенного извне поля СВЧ-диапазона. ЭПР наблюдается только от веществ, содержащих парамагнитные соединения. Возможности методов ЭПР-спектроскопии в последние годы значительно расширились за счет использования парамагнитных зондов для исследования соединений, не имеющих собственных молекул с неспаренными спинами электронов.

Применение методов ЭПР для тонких структурных исследований основано на том, что помимо приложенного извне постоянного магнитного поля, в котором наблюдается резонанс, в веществе существует внутреннее поле. Их влияние приводит к сдвигу резонанса, изменению формы резонансной линии (расщеплению, расширению), изменению интенсивности и т. д. Изучение этих характеристик позволяет судить не только о природе исследуемых веществ и последовательности расположения отдельных химических групп в молекулах, но и об их пространственной структуре, динамике движения отдельных молекулярных групп и молекул в целом. Очевидно, что применение методов парамагнитного резонанса открывает широкие возможности для изучения структуры биоло-