

ТЕХНИЧЕСКИЕ СРЕДСТВА АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА

УДК 681.327

Ю. Ю. ШТАТНОВ

(Новосибирск)

МЕТОДЫ ИСПРАВЛЕНИЯ ИНФОРМАЦИИ СРЕДСТВАМИ МАШИННОЙ ГРАФИКИ

1. **Постановка задачи.** Традиционные направления развития средств машинной графики затрагивают только одну сторону трехсторонней связи оператор — задача — машина [1—6]. Довольно часто встречаются ситуации, когда выходные данные одной задачи являются входными для другой. Во всех существующих системах вопрос кодирования и подготовки данных — дело рук самого пользователя, и этот процесс зачастую занимает больше времени, нежели сам счет. Данные для задач пользователя, будь то информация для расчетов или же какая-то деловая информация, как правило, содержат ошибки, на обнаружение которых затрачивается очень много времени и труда. Например, для решения задач теории упругости готовятся данные о координатах точек на контурах. Информация носит геометрический характер, однако поиск ошибок информации в числовом виде затруднен. Представив эту информацию в графическом виде, можно наглядно увидеть сделанные при вводе ошибки. Тем не менее исправление их в исходных данных все еще довольно сложно. В связи с этим возникает необходимость интерфейса между прикладными расчетными программами, где ведется обработка данных, и графической интерактивной системой как средства для обнаружения и корректировки ошибок данных. Смысл такого интерфейса заключается в том, что, исправляя ошибки в графической форме представления данных, тем самым исправляем ошибки в данных прикладной вычислительной задачи.

В связи с задачей создания такого интерфейса были разработаны общие принципы взаимодействия оператор — задача — машина. Для этого пользователь в своей задаче требуемые для расчетов данные представляет в графической форме. Возможность создания такой графической формы — назовем эти данные графическим образом — предоставляется пользователю средствами системы. После этого вступает в работу графический редактор системы и, таким образом, осуществляется традиционная связь оператор — графическая система. Для того чтобы обеспечить связь графическая система — прикладная программа, вводится протокол графических команд, выполняемых редактором во время своей работы. Одновременно с корректировкой изображения на экране ведется исправление графического образа данных согласно вводимым командам. Эту функцию также берет на себя система. После завершения работы редактора программе пользователя передается протокол графических команд, выполненных во время работы программы-редактора. При этом на пользователя ложится обязанность учета этих графических команд при корректировке исходной графической информации. Набор команд фиксирован, и поэтому для определенного класса задач пользователь должен написать один раз программы обработки графических команд.

Подводя итог, получаем цепочку: прикладная программа пользователя \Leftrightarrow графическая система \Leftrightarrow оператор. Система функционально может быть представлена в виде следующей цепочки: прикладная программа и исходные данные в ней \rightarrow создание графического образа исходных данных \rightarrow создание дисплейного файла \rightarrow ввод и исправление дисплейного файла программой «Графический редактор», ведение протокола выполненных графических команд \rightarrow исправление графического образа исходных данных по протоколу выполненных графических классов задач, решаемых в ИГД СО АН СССР:

- расчет напряжений в горных массивах;
- проектирование горных карьеров и подземных выработок;
- проектирование горных машин и механизмов.

Кроме вышеперечисленных функций, пользователю предоставляются некоторые сервисные возможности, а именно: ввод информации с кодировщика графической информации, а также графическое представление своих данных на графопостроителе. Причем ввод информации осуществляется в виде последовательности графических команд, а потом по этой последовательности создается файл графического образа. Это производится из тех соображений, что пользователь может быть сразу удовлетворен введенными данными и ему останется лишь обработать введенные команды для того, чтобы получить данные для своей задачи. В любом случае это косвенный ввод данных для задачи пользователя. Косвенным он является потому, что он должен обработать по установленным им самим правилам эти данные.

2. Графический интерфейс. Как уже упоминалось вначале, принцип обмена информацией между вычислительной задачей и графической системой заключается в том, что из программы пользователя в систему поступает графический образ исходной информации, а в программу пользователя из системы — файл графических команд, выполненных в результате работы графического редактора.

На основе анализа существующих в данной области работ было выбрано множество основных графических команд. Через это множество можно описать все остальные выполняемые команды. Интерфейс оператор — графическая система использует эти команды для того, чтобы выполнять необходимые изменения. Однако каждому действию оператора по исправлению изображения на экране графического дисплея обязательно соответствует одна графическая команда. Чаще всего это будет некоторая, строго определенная для каждого действия последовательность. При вводе новых режимов редактирования изображения на экране дисплея список графических команд не изменяется. Тем самым пользователь защищен от изменений в системе.

Набор базовых графических команд можно условно разделить на три группы команд: создания, исправления и листовых операций.

2.1. Команды создания. Эта группа команд предусмотрена в системе для того, чтобы пользователь мог вводить исходные данные не только в своей прикладной программе, но и с кодировщика графической информации. Система предоставляет пользователю определенный сервис, в том числе возможность ввода информации с кодировщика. Информация поступает в виде графических команд, которые впоследствии будут переданы в программу пользователя и обработаны его программами.

2.2. Команды исправления. Эта группа команд предназначена для отображения действий графического редактора. Каждому действию графического редактора ставится в соответствие некоторая последовательность графических команд. Например, действию в режиме редактирова-

ния «Удалить линию полностью» ставится в соответствие последовательность команд — «Определить номер текущей группы», «Разделить графический элемент на два», «Удалить графический элемент». Это в том случае, если элемент состоит из нескольких линий и не находится в данный момент в текущей группе.

2.3. Команды листовых операций. Ввиду того что лист является особой структурой, отличающейся от других графических элементов (см. п. 3), было решено выделить в особую группу геометрические операции над ними.

К группе листовых операций относятся: 1) разрезать лист; 2) найти объединение листов; 3) найти пересечение листов.

3. Структуры данных. Как уже описано выше, данная система предназначена для осуществления корректировки исходной математической информации графическими методами. Для этой цели создаются следующие файлы данных:

- 1) файл графического образа исходной информации;
- 2) файл протокола графических команд, который называется файлом графического протокола;
- 3) дисплейный файл.

Файл графического образа содержит структурированный образ исходной информации в графическом виде. Как правило, он создается пользователем в своей прикладной программе и передается графической системе. Кроме того, образ может быть создан в конце работы графической системы по имеющемуся файлу графических команд. Этот вариант имеет место в том случае, когда ввод информации осуществляется с кодировщика. По окончании работы графической системы он корректируется и записывается на диск или ленту. Система предоставляет также дополнительный сервис для пользователя: графический образ или его часть можно построить на графопостроителе, получив тем самым твердую копию графического образа исходной информации. Другого доступа к графическому образу пользователь не имеет.

Файл графических команд содержит протокол графического взаимодействия пользователя и графической системы. Он может быть создан на этапе ввода с кодировщика графической информации. При кодировании с кодировщика идет последовательность команд создания, которые записываются в файл графического протокола. Кроме того, этот файл может быть создан в результате работы пользователя с графическим редактором данной системы. Графический протокол после окончания работы передается в прикладную программу пользователя, которому система предоставляет возможность последовательного чтения из него графических команд.

Под элементом в системе понимается одна из следующих структур: 1) точка; 2) линия незамкнутая; 3) линия замкнутая; 4) лист; 5) текст (строка текста).

Структуры данных типа 1—4 называются графическими элементами. Графический элемент может состоять из одного или нескольких множеств точек, которые кодируются по своим абсолютным координатам относительно центра системы координат, где создается графический образ исходной математической информации. Листом называется двумерная или трехмерная область, ограниченная набором замкнутых линий и в некоторых случаях основной рамкой. Первая замкнутая линия листа кодируется одним множеством точек и обходом против часовой стрелки, остальные — обходом по часовой стрелке. Замкнутая линия кодируется одним множеством точек и обходом против часовой стрелки.

Заключение. Принципы, послужившие основой для создания этой системы, были изложены в работе [7]. Для случая, когда исходной информацией является графическая информация, создана программная реализация данной системы на ЭВМ СМ-4 [8]. В настоящий момент система применяется к указанному в п. 2 классу вычислительных задач. Однако возможность применения этой системы к указанному классу задач не

исчерпывается. Благодаря вышеперечисленным достоинствам система достаточно универсальна и может быть применена для других классов задач.

ЛИТЕРАТУРА

1. Баяковский Ю. М., Михайлова Т. Н., Мишакова С. Т. ГРАФОР — комплекс графических программ на Фортране.— М., 1972. (Препринт/АН СССР, ИПМ; 41).
2. Математическое обеспечение графопостроителей. СМОГ, 1 уровень: Инструкция по программированию/Под ред. Ю. А. Кузнецова.— Новосибирск: ВЦ СО АН СССР, 1976.
3. Энджел И. Практическое введение в машинную графику.— М.: Радио и связь, 1984.
4. Пилюгин В. В., Сумароков Л. Н., Фролов К. В. Машинная графика и автоматизация научных исследований // Вестн. АН СССР.— 1985.— № 10.
5. Гиллой В. Интерактивная машинная графика.— М.: Мир, 1981.
6. Фоли Дж., Вэн Дэм А. Основы интерактивной машинной графики.— Т. 1—2.— М.: Мир, 1985.
7. Штатнов Ю. Ю. Редактирование графической и математической информации на ЭВМ. Институт горного дела СО АН СССР.— Новосибирск, 1981.— Рукопись деп. в ВИНТИ, № 4391—81.
8. Штатнов Ю. Ю. К вопросу о редактировании математической информации с помощью графических методов. Институт горного дела СО АН СССР.— Новосибирск, 1984.— Рукопись деп. в ВИНТИ, № 6206—84.

Поступила в редакцию 14 ноября 1986 г.

УДК 681.3.06

П. Л. ХРАПКИН

(Новосибирск)

ОБЗОР СРЕДСТВ ПРОГРАММИРОВАНИЯ АППАРАТУРЫ КАМАК

За годы применения стандарта КАМАК в нашей стране сформировался практический набор систем программирования такой аппаратуры. Цель этой статьи — не только рассмотреть многообразие наиболее распространенных средств программирования КАМАК, но и отразить стремление их к унификации, выявить наиболее удобные для массового применения. Предложено типовое решение, выбор которого основан на опыте по применению мини- и микроЭВМ серии СМ и «Электроника», накопленном в Институте ядерной физики СО АН СССР. То же стремление к унификации программных средств систем автоматизации, использующих аппаратуру КАМАК, выражено в ряде обзорных работ [1, 2 и др.].

Уже в самых первых системах автоматизации в ИЯФ СО АН СССР, использующих аппаратуру КАМАК, сложилась практика составления управляющих программ на Фортране. Для работы с конкретными модулями КАМАК используются соответствующие подпрограммы, написанные на языке уровня ассемблера или на Фортране. В последнем случае ассемблер используется лишь для базовых универсальных подпрограмм, ориентированных на работу с любым устройством КАМАК [3].

Несмотря на значительные успехи в области разработки языков программирования, связанные с созданием таких мощных современных языков, как Паскаль, Си и др., наиболее распространен при создании систем автоматизации большого и среднего размеров, казалось бы, давно устаревший Фортран. Конечно, небольшой алгоритм размером в несколько десятков или даже сотен строк можно запрограммировать на ассемблере значительно эффективнее, чем на Фортране, однако пользователь,