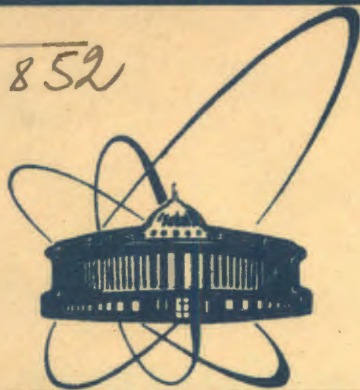


C-852



сообщения
объединенного
института
ядерных
исследований
Дубна

3661/2-81

20/r11-81

P10-81-360



Т.А.Стриж

РЕЖИМ ДИАЛОГА В СИСТЕМЕ ГЕНЕРАЦИИ
ПРОГРАММ ОБРАБОТКИ
ФИЛЬМОВОЙ ИНФОРМАЦИИ

1981

Оснащение ЭВМ CDC-6500 в ОИЯИ терминальными устройствами и соответствующим математическим обеспечением (подсистемой INTERCOM /1/) создало хорошие условия для создания специализированных систем, использующих режим диалога.

На базе разработанной методики применения этого режима в системе математической обработки फिल्मовой информации в ОИЯИ создан ряд программ /2/.

В данном сообщении рассматривается организация диалогового режима работы при генерации программ обработки फिल्मовой информации /3/.

1. Система генерации программ обработки данных камерных экспериментов

Система генерации построена на базе стандартного математического обеспечения с широким использованием имеющегося программного аппарата. Она служит для получения различных версий программ обработки फिल्मовой информации в ОИЯИ. В системе обеспечена возможность изменения значений любых констант, использующихся при обсчете данных.

Генерация программ осуществляется в два этапа. На первом по запросу пользователя составляется задание программе-сборщику системы. Проводится синтаксический и семантический анализ запросов. На втором - программа-сборщик собирает текст программы из текстовых фрагментов, находящихся в специальных банках текстовых фрагментов (РАМ-файлах). Собранная программа затем передается на трансляцию и выполнение.

Пакетный режим обработки, использующийся в системе генерации, не требует дополнительных затрат ни ресурсов ЭВМ на ведение диалога, ни времени пользователя, который должен находиться у терминального устройства в процессе работы. При этом полностью сформулированное задание на входном языке системы включается во входной поток задач на ЭВМ.

Однако ошибки в задании пользователя, либо неполное описание программы влекут за собой прекращение работы и требуют повторного

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

обращения к системе генерации. Пакетный режим обработки также лишает пользователя возможности оперативно вносить дополнения и изменения в описание требуемого варианта программы.

2. Возможности диалога в системе генерации

Диалоговый режим лишен ограничений, перечисленных выше.

На одной из базовых ЭВМ ОИЯИ СДС-6500 диалог реализуется через подсистему ИНТЕРСОМ ^{/1/}. Использование имеющихся технических и программных средств накладывает определенные ограничения на диалоговые системы:

- во-первых, объем оперативной памяти ЭВМ СДС-6500, требующийся для их работы, не должен превышать 24-30 тыс. слов;
- во-вторых, время выполнения программы в центральном процессоре или ряда программ, использующих диалог, должно быть не более 320 с.

Ввиду того, что система генерации рассчитана на создание большого числа разнообразных программ и ориентирована на широкий круг пользователей, в ней удобно использовать режим сопровождения на основе директивного языка ^{/4,5,10/}. Это позволяет получать заведомо синтаксически правильное описание требуемого варианта программы. Режим сопровождения также обеспечивает получение нужных промежуточных результатов после завершения заданных этапов работы системы.

Использование директивного языка позволяет формулировать запросы от системы генерации в виде развернутых сообщений. Ответы пользователя формируются в виде команд-директив, позволяющих выполнять широкий круг операций.

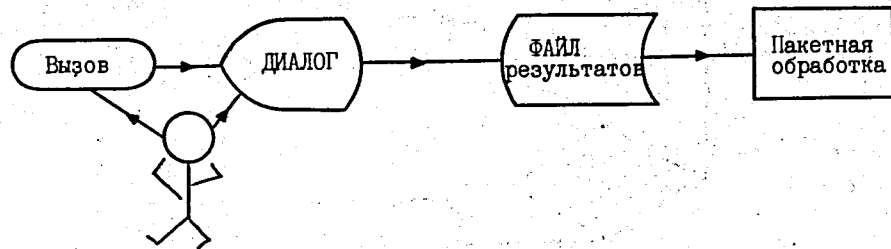
3. Организация режима диалога

В связи с тем, что при работе ЭВМ в реальном времени необходимо обеспечить время ответа от 2 до 15 с ^{/8/}, на длину программ и время счета накладываются ограничения, приведенные выше.

Поэтому режим диалога целесообразно использовать для решения небольших задач, не требующих значительного времени и памяти ЭВМ.

Система генерации, разработанная в ОИЯИ, используется для сборки программ обработки फिल्मовой информации, которые относятся к классу "больших" программ. Они требуют большого объема памяти и чаще всего используются для обсчета значительных массивов данных в одном сеансе. Сборка текста этих программ и трансляция требуют ~ 20-25 с и ~ 100-120 с времени центрального процессора соответственно.

Эта специфика класса генерируемых программ привела к использованию следующей схемы работы в системе, включающей как режим диалога, так и пакетную обработку.



Таким образом диалог в системе генерации используется при формировании требуемой программы, а остальные операции с ней (сборка, трансляция и т.п.), требующие как большого времени, так и объема памяти ЭВМ, выполняются в пакетном режиме.

Остановимся подробно на организации диалога в системе. Поскольку генерация программ рассчитана на широкий круг пользователей, не имеющих хорошей подготовки для работы на ЭВМ, то диалог в ней строится по принципу обучающе-решающих систем ^{/7/}.

В этом случае система знакомит пользователя с имеющимися возможностями и указывает ему способ выбора нужного варианта и контролирует его работу. При обнаружении ошибок в полученном от пользователя задании система сообщает ему об этом и "подсказывает" способ устранения ошибки. На каждый ответ пользователя выдается определенная информация, и так до тех пор, пока задание не составлено полностью. После этого оно передается для выполнения, а результаты записываются на специальный файл.

Организация режима диалога на этом этапе показана на рис. 1. Диалог здесь:

- обеспечивает связь с пользователем;
- снабжает пользователей сведениями информационно-обучающего характера;
- осуществляет диагностику ошибок, сделанных пользователями при вводе управляющей информации;
- формирует и накапливает в соответствии с указаниями пользователя директивы языка запросов для сборки нужной ему версии программы;
- переводит задания на языке запросов на язык директив программы-редактора УРАТСКУ ;
- передает сформированное задание на счет в режиме пакетной обработки.

Программы обработки फिल्मовой информации имеют некоторые особенности, которые находят свое отражение в разработке диалога для состав-

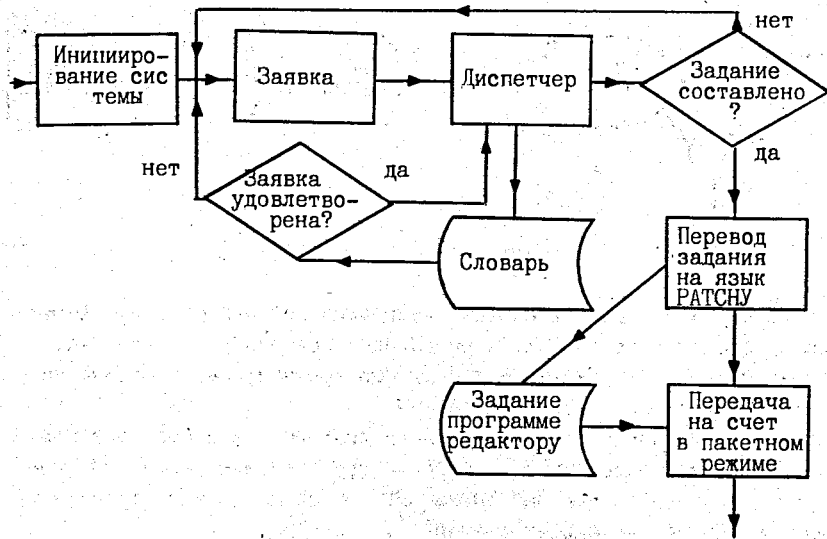


Рис. 1

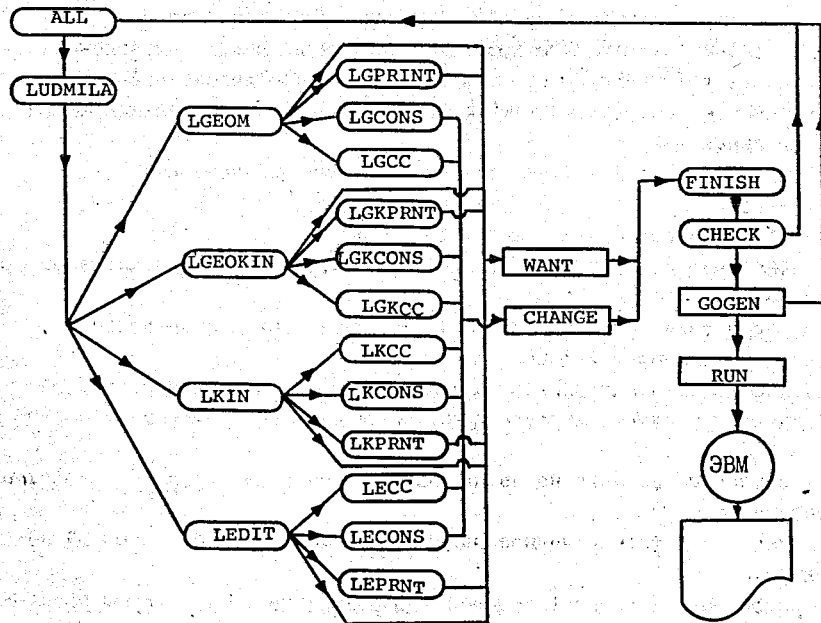


Рис. 2

ления задания на генерацию таких программ. На рис. 3 приведена упрощенная схема процесса выбора нужной версии программы.

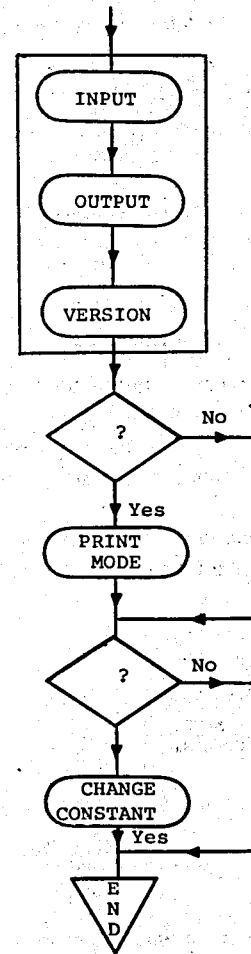


Рис. 3.

В соответствии с этой схемой диалог должен обеспечить:

- выбор соответствующей версии программы, определение типа входных данных и способа записи результатов;
- выбор требуемой формы представления результатов счета;
- выбор и замену значений соответствующих констант, используемых при счете.

Этим определяется введение в диалог специальных напоминающих-предписывающих инструкций в виде мнемонических кодов для описания различных версий программ, констант и их значений.

Заклочительным этапом работы системы генерации в диалоговом режиме является передача задания на дальнейшую сборку программы в пакетном режиме.

Для этих целей после составления задания редактору уРАТЧУ система генерации выдает пользователю запрос на данные, необходимые для перехода в пакетный режим.

Такое разделение функций между диалоговой и пакетной обработкой связано, в основном, с затратами времени центрального процессора на осуществление основных этапов генерации. Так, для составления описания требующейся программы, содержащей около 25000 операторов, требуется ~ 1 с, на синтаксический и семантический анализ задания ~ 1,5 с, что вполне приемлемо для диалога [8].

Время работы центрального процессора при генерации в интерактивном режиме с использованием обучения ~ 8 с. Время работы за терминалом ~ 12 мин.

4. Язык диалога

Диалог между ЭВМ и пользователем в системе генерации ведется на вполне понятном языке с использованием хорошо известных терминов для описания возможностей системы. Ниже приводится описание директив языка диалога с помощью металингвистических формул:

```
<директива> ::= <имя директивы> <признак конца директивы> |
<имя директивы> (< список параметров > ) |
<напоминающая-предписывающая инструкция>
<имя директивы> ::= ALL/EXP/WANT/CHANGE/ARRAY/UNIT/CHECK/COGEN/
RUN/COLLECT/PFTN/LIB
<признак конца директивы> ::= .
<список параметров> ::= < параметр > | < список параметров > , < параметр >
<параметр> ::= < пусто > | < значение параметра >
< пусто > ::=
< значение параметра > ::= < БЦ символ > ↑ 40
< БЦ символ > ::= < буква > | < цифра >
< буква > ::= A/B/C/D.../Z
< цифра > ::= 0/1/2/3/4/5/6/7/8/9
<напоминающая-предписывающая инструкция> ::= < БЦ символ > ↑ 7
```

Обозначение ↑ n указывает на употребление указанного до него символа от 1 до n раз. Синтаксис языка диалога довольно прост. Анализ в основном организуется для тех директив, которые состоят из имени и параметров. При этом контролируется правильность задания параметров, полнота всего списка параметров.

Ниже приведен пример диалога в системе генерации. Подчеркнутые директивы вводятся пользователем, остальные являются ответами системы генерации.

EXP, LUDMILA

LUDMILA GEOMETRY - VERSIONS AVAILABLE

```
* VERSION *
LUDMILA GEOMETRY          * NOT MIX WITH OVERLAY GEOMETRY
LUDMILA OVERLAY GEOMETRY  * NOT MIX WITH          GEOMETRY
  INPUT HPD
  INPUT PUOS
  OUTPUT IN THRESH FORMAT
  OUTPUT IN GRIND CONVERSION
  HISTOGRAMS FOR LUDMILA
  EXP 207 LUDMILA
```

SELECT A VERSION NAME FROM THE ABOVE LIST AND TYPE

WANT, * VERSION .

EXAMPLE WANT, LUDMILA GEOMETRY, INPUT HPD, OUTPUT IN THRESH FORMAT.

*** WARNING: SOME VERSIONS SHOULD NOT BE MIXED

*** NOTATION 1 OUTPUT PRINT

IF YOU WANT SAME PRINT OUTPUT TYPE

LGPRINT.

*** NOTATION 2 CONSTANTS VALUE CHANGING

IF YOU WANT CHANGE SOME CONSTANTS VALUES, TYPE

LGCONST.

*** NOTATION 3 IF YOUR WORK IS FINISHED TYPE FINISH.

WANT, LUDMILA GEOMETRY, INPUT HPD, OUTPUT IN THRESH FORMAT.

LGPRINT.

FIDUCIAL AND VERTEX RECONSTRUCTION
PRINT MODES - SET AVAILABLE

```
PRINT FIDUCIAL AND VERTEX RECONSTRUCTION
PRINT STEERING FOR FINAL VERTEX FIT
PRINT FINAL VERTEX FIT
PRINT FIDUCIAL COEFFICIENTS COMPUTATION
PRINT POINT FIT
PRINT STEERING LABELLED POINT RECONSTRUCTION
PRINT POINT MATCH STEERING
PRINT POINT OR TRACK MATCH DECISION
PRINT 2POINT TRACK RECONSTRUCTION
PRINT 2POINT SETUP
DEBUG FIDUCIAL AND VERTEX RECONSTRUCTION
DEBUG FINAL VERTEX FIT
DEBUG FIDUCIAL COEFFICIENTS COMPUTATION
DEBUG SPACE POINT APPROXIMATION FULL LABELLING
DEBUG POINT MULTIPLES
DEBUG POINT OR TRACK SEARCH
```

SELECT PRINT MODE AND TYPE
WANT, PRINT MODE.

EXAMPLE: WANT, DEBUG POINT MULTIPLES, PRINT 2POINT SETUP.
IF YOU WORK IS AT END TYPE FINISH.

WANT, PRINT 2POINT SETUP.

LGCONST.

CONSTANTS THAT CAN BE CHANGED - SET AVAILABLE

TMEASR /0.017	TRACK SETTING ERROR
TMSCON /0.0005	MULTIPLE SCATTERING CONSTANT
DLENDP /0.1	ERROR ON LENGTH FOR TRACK WITH END POINT
DMAXFD /0.04	MAXIMUM RESIDUALS FOR FIDUCIALS
REFSTX /0.0	EXPECTED SQUARED STRETCH
TAMAX /0.45	SINE OF MAXIMUM HALF TURNING ANGLE
TLMAX /60.	MAXIMUM TRACK LENGTH CUT OFF
NCFMAX /25.	MAX NO. OF NEAR CORRESPONDING POINTS WHICH MAY BE STORED IN SC2 BANK WITHOUT TRUNCATION
DSMAX /80.	ABSOLUTE MAXIMUM ARC LENGTH
TVDEV /0.3	MAX DISTANCE FROM VERTEX TO CIRCLE FIT
EAMAX /0.03	MAX ERROR ON FITTED TRACK ANGLE AT VERTEX
ECMAX /0.003	MAXIMUM ERROR ON FITTED CURVATURE

CHANGE CONSTANTS FROM THE ABOVE LIST BY TYPING

CHANGE, CONSTNAME, \$CONSTOLDVALUE\$, \$CONSTNEWVALUE\$.

***** EXAMPLE: CHANGE, AXINV, \$-1.\$, \$1.\$, DSMAX, \$80.\$, \$50.\$.

NB THE SIGN \$ IS VERY IMPORTANT ***** NB***

IF YOUR WORK ARE AT END TYPE FINISH.

IF YOU WANT CHANGE ANY OTHER CONATANTS TYPE

LGCC.

CHANGE, EAMAX, \$0.0.3\$, \$0.005\$.

FINISH.

IF YOUR WORK IS FINISHED TYPE CHECK.

CHECK.

YOU ENTERED NEXT REQUEST DECK, CHECK IT PLEASE

LUDMILA GEOMETRY
INPUT HPD
OUTPUT IN THRESH FORMAT
PRINT 2POINT SETUP
EAMAX/0.03

EAMAX/0.005

IF YOUR REQUEST DECK IS O.K. TYPE GOGEN,*NAME*, PFN, ID.
WHERE *NAME* IS NAME FROM FOLLOWING LIST

LUDMILA
RISK
VPK100
MIS

AND PFN - YOUR PERMANENT FILE NAME TO CATALOG THE CRADLE
ID - YOUR PFN USER-ID

IF YOUR REQUEST DECK IS NOT O.K. TRY AGAIN. TYPE ALL AND SO ON

GOGEN, LUDMILA, CRAD, LCTIVA.

NOW YOU HAVE A PERMANENT FILE WITH NAME CRAD AND ID=LCTIVA.

YOU MUST USE IT IN PROCEDURE RUN

WHEN YOU SEND JOB IN CENTRAL SITE

PROCEDURE RUN IS CALLED IN FOLLOWING WAY

RUN, JOB, TIME, PASS, EXPERIMENT, DISK, VSN, SN, PFN, USER-ID, DATA, ID-DATA
WHERE

JOBN	USER JOB CARD NAME
TIME	TIME LIMIT ON JOB CARD
PASS	USER PASS ON ACCOUNT CARD
EXPERIMENT	EXPERIMENT NAME (LUDMILA, RISK, VPK100, MIS)
DISK	IF DATA ON DISK TYPE 1, IF ON TAPE = 0
VSN	DISK OR TAPE VSN
SN	DISK SETNAME
PFN	CRAD FOR YOU JOB
USER-ID	LCTIVA FOR YOU JOB
DATA	FILE NAME WITH DATA
DATA-ID	DATA FILE ID

***** EXAMPLE:

RUN, CSTR2, T200, PLCTA, LUDMILA, 1, 106, LOBIVI, CRAD, LCTIVA, HPDDATA, LCTAI.

RUN, MYNAME, T300, MYLAB, LUDMILA, 1, 106, LOBIVI, CRAD, LCTIVA,
HPDDATA, LVECR.

Стандартной реакцией на ошибки является распечатка ошибочного параметра с соответствующей диагностикой.

Широкое использование для реализации режима диалога возможностей языка SCL /9/ значительно упростило процедуру анализа директив. Командное слово каждой директивы и напоминающая-предписывающая инструкция в виде мнемонического кода являются названиями соответствующих процедур на языке SCL. Все эти процедуры находятся на специальном библиотечном файле, выполняющем роль диспетчера системы. Ошибка в командном слове опознается на уровне языка SCL. После диагностики об отсутствии заявленной процедуры (команды) необходимо повторить команду без грамматических ошибок. Организация диалога посредством языка SCL позволяет возвращаться назад на любое число шагов при составлении задания, повторять директивы, начинать составление задания вновь.

На рис.2 приведена блок-схема процедур языка SCL, реализующих генерацию программ для экспериментов, проводимых на камере "Людмила". Указаны командные слова и мнемонические коды соответствующих процедур.

Заключение

Благодаря широкому использованию базового математического обеспечения ЭМ СРС-6500 (подсистемы INTERCOM и языка SCL) и проблемно-ориентированного математического обеспечения (система генерации программ обработки фильмовой информации) в короткий срок было создано программное обеспечение режима диалога в системе генерации.

Понятный физикам язык, совмещение информационного и обучающего обеспечения системы генерации сделали последнюю доступной широкому кругу пользователей (практически без потерь времени на ее освоение). При этом обеспечивается активное участие пользователя в процессе составления задания в условиях надежного контроля его действий со стороны системы.

В заключение автор пользуется случаем выразить глубокую благодарность Н.Н.Говоруно за поддержку работы, В.Г.Иванову за поддержку, постоянную помощь и внимание; С.Г.Бадаляну, А.Дирнеру, Н.А.Буздавиной, И.И.Шелонцеву, О.В.Благодирской за полезные обсуждения и дружескую критику, А.А.Корнейчуку за ряд ценных замечаний.

Литература

1. INTERCOM Reference Manual. Control Data Corp. Publ. 60307100, USA, 1974.
2. Бадалян С.Г. и др. ОИЯИ, P10-12744, Дубна, 1979.
Бадалян С.Г. ОИЯИ, P10-12879, Дубна, 1979.
3. Говорун Н.Н. и др. ОИЯИ, P10-11612, Дубна, 1978.
Бадалян С.Г. и др. ОИЯИ, P10-11911, Дубна, 1978.
4. Гурова Л.И., Сахаров С.С. Прикладные программы. "Статистика", М., 1980.
5. Нахмансон М.С. и др. УСИМ, 1980, № 5, с.111.
6. Бадалян С.Г. ОИЯИ, P10-12880, Дубна, 1979.
7. Глушков В.М. УСИМ, 1974, № 1, с.3.
8. Мартин Дж. Системный анализ передачи данных. "Мир", М., 1975, т.1.
9. CDC NOS/VE 1 Reference Manual, Pub. No. 60493800, Control Data Corporation, St. Paul, Minnesota, USA.
10. Брановицкий В.И. и др. УСИМ, 1979 № 5, с.18.
11. Хигман Б. Сравнительное изучение языков программирования. "Мир", М., 1974.

Рукопись поступила в издательский отдел
29 мая 1981 года.