

УДК 681.142.1

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРИКЛАДНЫХ НАУЧНЫХ ИССЛЕДОВАНИЙ

В.Т.Еркаев, В.И.Константинов, Д.И.Свириденко, Н.Т.Тукубаев

Создание алгоритмов решения прикладных задач на ЭВМ является в настоящее время одним из основных и важнейших видов деятельности — в прикладной математике. Одновременно растет роль машинного эксперимента и в самой математике [6]. Законченные исследования в данном направлении обычно завершаются разработкой систем программ. В связи с этим актуальной проблемой прикладной математики является перевод алгоритмического фонда в программный фонд, его сопровождение и эксплуатация. Данная проблема породила особый вид деятельности по машинной реализации прикладных научных исследований. Если на начальных стадиях развития вычислительной техники программный продукт был побочным продуктом разработчиков алгоритмов, то в настоящее время деятельность по программному обеспечению в большинстве случаев уже считается профессиональной. Результат этой деятельности — специальные программные системы — оформляются в виде пакетов прикладных программ.

Прикладное программирование в настоящее время становится одним из самых быстроразвивающихся направлений прикладной математики^{*)}. Вместе с ростом значения программного обеспечения возрастает роль методологических и технологических исследований в данной области. Рассмотрению некоторых методологических и технологических вопросов разработки программных комплексов для решения научно-прикладных задач на ЕС ЭВМ и посвящена настоящая работа.

*) Так, например, в США ежегодный прирост в области программного обеспечения составляет около 27% [8].

§1. Конструирование пакетов программ

Текущий этап развития прикладного программного обеспечения характеризуется определенным усложнением деятельности ^{ж)} в данной области, обусловленным:

- растущей сложностью решаемых задач, их комплексным характером;

- возросшими требованиями к программному продукту ^{жз)}, что, например, привело к развитию архитектуры прикладных программных систем от простого набора (библиотеки) функциональных подпрограмм до пакетов программ;

- сложностью и большими возможностями современных ЭВМ и их системного математического обеспечения, что особенно характерно для ЕС ЭВМ с многотомными описаниями ОС ЕС. Для активного общения с ЭВМ, грамотной организации и "погружения" программной продукции в среду ОС ЕС требуется определенный профессионализм, которого, как правило, нет у специалистов-прикладников, имеющих программистскую подготовку в рамках традиционных языков прикладного программирования (АЛГОЛ, ФОРТРАН, КОБОЛ и т.п.);

- некоторым отдалением ЭВМ от прикладных программистов в процессе развития операционных систем пакетной обработки.

С другой стороны, высокие темпы развития системного математического обеспечения ЕС ЭВМ, сменяемость поколений ЭВМ требуют соответствующих темпов разработки прикладного программного обеспечения. Однако на практике прикладное программирование отстает от разработки системного математического обеспечения, что обусловлено отмеченными выше причинами. Поэтому естествен интерес прикладных программистов к системам, автоматизирующим процесс конструирования пакетов прикладных программ и сокращающим сроки их разработки. В основе таких систем лежит идея внедрения индустриальных методов в процесс изготовления программных изделий. Заметим, что возможность стандартизации архитектуры пакетов прикладных программ и определенная инвариантность их системного наполнения относительно функционального являются хорошей базой для реал-

ж) В дальнейшем эту деятельность мы будем называть конструированием пакетов программ.

жз) Низкая стоимость, высокое качество, гарантированная работоспособность, доступность пользователям-непрофессионалам, небольшие затраты на ввод и обслуживание, хорошая документация, возможность модификации и развития и т.п.

лизации этой идеи. Широко используется, например, метод наполнения различным функциональным содержанием некоторой метасистемы, содержащей готовую системную часть пакета [3-5].

Индустриализация конструирования пакетов программы предполагает наличие:

- определенной методологии конструирования, направленной на стандартизацию архитектуры программных изделий;
- развитой технологии программирования, основанной на модульном изготовлении программных изделий, эффективных индустриальных системах программирования;
- четкой организации деятельности коллективов разработчиков и гибкого управления технологическим процессом изготовления программных изделий.

В настоящее время имеется ряд достаточно универсальных технологий программирования: технология структурированного программирования [9], R-технология [7], технология расслоенного программирования [10] - обеспеченных соответствующими инструментальными комплексами. Тем не менее ни одна из них не покрывает полностью многоаспектной деятельности по конструированию пакетов прикладных программ^{*)}. Для решения вопросов, связанных с индустриализацией конструирования программного обеспечения, в настоящей работе предлагается концепция технологического комплекса, понимаемого как взаимоувязанная совокупность инструментально обеспеченных технологий для более широкого спектра работ по конструированию пакетов прикладных программ.

В технологический комплекс предполагается включить эффективную для каждого вида работ технологию, подобранную из числа имеющихся или же специально разработанную. Кроме того, предполагается учесть следующие особенности разработки программных систем в академических учреждениях:

- работа, как правило, ведется малыми коллективами;
- ощущается дефицит хороших системных программистов;
- отсутствуют стандарты на изготовление и документирование продукции прикладного научного программирования;

*) Эта деятельность включает проектирование, программирование модулей, конструирование сложных управляющих программ, лингвистическое обеспечение, производство системных программ, связь с ОС ЭЕМ, отладку, документирование и передачу потребителям программных изделий и т.п.

- активно используется собственная программная продукция;
- для большинства решаемых задач характерны простые (регулярные) структуры данных и сложные алгоритмы их обработки;
- широко используется диалоговый режим для экспериментирования на ЭВМ, поиска и оптимизации стратегий решения задач.

Ниже излагается один из подходов к созданию такого комплекса на базе ОС ЕС ЭВМ.

§2. Технологический комплекс - система ТЕКОМ - конструирования прикладных программных систем

Система ТЕКОМ, исходя из определенной методологии конструирования пакетов прикладных программ, призвана обеспечить рациональную организацию работ по созданию программных изделий на основе индустриальных методов.

Предлагается вести изготовление программных изделий на основе развивающегося банка модулей, имеющего иерархическую трехуровневую структуру. Модулем назовем некоторую конструктивную единицу организации программной продукции, ее унифицированный элемент, являющийся составной частью программной системы. Каждый уровень банка модулей составляет модули одного типа, образующие фонд, пополняемый за счет вновь создаваемых, заимствованных и модифицируемых модулей. В системе ТЕКОМ унифицированы правила и средства композиции модулей более высокого уровня. Архитектура модулей ориентирована на реализацию в среде ОС ЕС ЭВМ. Не вдаваясь в подробную спецификацию, кратко охарактеризуем предлагаемые типы модулей.

На первом, низшем уровне банка модулей находятся элементарные модули "процедуры". Это программы, обрабатывающие объект в оперативной памяти. Они не содержат операторов ввода-вывода, что позволяет легко добавлять или изымать такие модули из любой сложной программы. Второй уровень образуют модули-агрегаты ("подпрограммы" и "программы"), являющиеся в общем случае сложными, многовариантными программными изделиями, реализующими некоторый цикл обработки информации, расположенной на внешних устройствах (ввод-обработка-вывод). Выполнение "программы" является шагом задания для ОС ЕС. В рамках программы осуществляется комплексация модулей первого уровня; главный (управляющий) модуль программы является специализированным монитором, реализующим логическую схему реше-

ний некоторой задачи с использованием стандартного сервиса РЕДАКТОРА СВЯЗЕЙ ОС. Информационный интерфейс между модулями программы осуществляется в оперативной памяти. Комплексация программ для организации решения сложных задач целесообразно осуществлять на языке управления заданиями (JCL) ОС ЕС. Для этого над базисом имеющейся совокупности системных и проблемных программ строится лингвистическая надстройка в виде языка общения, отражающего функциональные возможности программы и их комбинаций. Монитором является программа, транслирующая инструкции языка общения в программу на языке JCL, выполнение которой осуществляет ОС ЕС. Для системных обслуживающих программ, наиболее часто используемых при работе на ЭВМ, уместно, кроме этого, обеспечение оперативного их выполнения путем динамической загрузки. Монитор должен обеспечить работу с программным фондом на кратком, но достаточно естественном для пользователей языке как в пакетном, так и в диалоговом режиме, развивая аппарат каталогизированных процедур и используя их в качестве JCL-макросов.

Краткость директив языка общения, позволяющая работать на ЭВМ в режиме диалога, достигается сокращенной мнемоникой, отказом от чрезмерного разнообразия возможностей системных программ за счет фиксации некоторых параметров в JCL-процедурах, стандартизации имен системных библиотек и хранения сведений о наборах данных в системном каталоге.

Таким образом, системные обслуживающие программы ОС сопрягаются с монитором, образуя мониторную систему, в значительной мере удовлетворяющую системным требованиям к пакету прикладных программ за счет средств операционной системы.

В результате эксплуатации программного фонда можно на определенном этапе выделить совокупность прикладных программ, управляемых мониторной системой и "покрывающих" с достаточной полнотой некоторую предметную область. Такую совокупность программ, оформленную вместе с мониторной системой для передачи другим пользователям как единое готовое программное изделие, назовем пакетом программ, а процесс его комплектации — генерацией пакета. Пакеты программ образуют третий уровень банка модулей. Межпрограммный информационный интерфейс в пакетах осуществляется во внешней памяти ЭВМ. Банк модулей, обслуживаемый технологическим комплексом, можно считать метасистемой, а пакеты программ — модулями этой метасистемы, рассчитанными на автономное использование. Такие пакеты естественно совместимы и могут быть легко объединены.

Будем различать два типа использования программного изделия: пассивное и активное. П а с с и в н о е и с п о л ь з о в а - н и е предполагает отношение к программному изделию как к готовому изделию, с эксплуатацией всех его возможностей без знания внутреннего устройства и без самостоятельной модификации программного изделия, хотя фонд модулей может быть открыт для автономного использования.

При а к т и в н о м и с п о л ь з о в а н и и пользователь может модифицировать программное изделие, строить собственные стратегии использования программы, изменять язык общения и т.п. Наиболее активными пользователями являются обычно сами разработчики программной системы. Поэтому в областях, где преобладает активный характер использования пакетов прикладных программ [4], к ним предъявляются те же требования, что и к метасистеме.

Таким образом, если пассивный пользователь нуждается в пакете программы, то активный - в самой метасистеме. Выделение уровней банка модулей является, на наш взгляд, рациональной декомпозицией процесса производства пакета прикладных программ. Каждый уровень банка определяет работу по производству программного изделия соответствующего типа, которая имеет свои особенности, требует своих инструментальных средств и своей технологии. Под технологией некоторой работы подразумевается частично упорядоченная последовательность операций (операционный маршрут), регламентирующих эту работу и приводящих к ее выполнению. На каждом уровне банка выполняется одинаковый перечень видов работ, но над программными изделиями различных типов: проектирование, программирование, подготовка тестирования, отладка, документирование, включение в банк модулей, передача пользователям, сопровождение. ТЕКОМ призван обеспечить пользователей технологическими схемами выполнения этих работ. Ясно, что, как и любые схемы, они будут конкретизированы при разработке конкретного программного изделия.

Важнейшими организационными принципами системы ТЕКОМ являются распараллеливание работ и специализация подразделений. Особенностью предлагаемой схемы построения банка модулей является возможность распараллеливания работ на трех уровнях: изготовления процедур, конструирования программы и их пакетирования, со специализацией подразделений, выполняющих эти работы, чему способствует различие инструментальных средств их выполнения. С другой стороны, на каждом уровне возможно и необходимо вести параллельно не-

которые технологические операции, такие, например, как кодирование, подготовка к тестированию и документирование программного изделия. Это создает предпосылки для специализации разработчиков внутри подразделений по видам работ и организации бригадного метода изготовления программного изделия.

Основой обеспечения организации и управления производством программных изделий является информационная модель процесса их изготовления, подобная модели основного производства в АСУ "Сигма" [11]. Для каждого изделия в модели хранится технология (операционный маршрут) его изготовления с закреплением операций за конкретными исполнителями, плановые и фактические данные о разработке, данные о взаимосвязи программных изделий друг с другом. Совокупность таких моделей для всех находящихся в разработке программных изделий составляет информационную модель проекта. Создание и поддержка актуального состояния этой модели может вестись на ЭВМ (для очень больших программных систем) или вручную, на стандартных бланках. К этой информации обеспечивается доступ всех разработчиков, и на ее основе производится планирование, учет, анализ хода разработки и взаимодействие разработчиков.

Рассмотрим некоторые принципы технологии документирования в системе ТЕКОМ. Документирование должно сопровождать любые стадии и виды работ и отражать результаты этих работ в текстовальном виде.

Предполагается два типа документации: рабочая документация, которая сопровождает процесс производства программных изделий и содержит самые подробные данные о нем, и документация готового программного изделия, которая составляется на основании рабочей и предназначается его потребителю. Конкретный состав документов и распределение информации по документам должны удовлетворять требованиям удобства формирования этой документации разработчиками (технологичность документов), с одной стороны, и удобства использования различными категориями пользователей — с другой. В идеале вся документация должна вестись на ЭВМ; с учетом высокой пока стоимости подготовки и хранения текстовой информации и недостаточности объемов внешней памяти прямого доступа, возможно частичное машинное ведение документации. Очень важно вести документирование параллельно с изготовлением программного изделия, широко используя принцип самодokumentирования программ (мнемоничность имен, структурированность,

комментарии, хороший формат распечатки). Стандарт документирования определяет структуру и состав документов. Различные системы стандартов отличаются формами представления одного и того же содержания. Машинное ведение документации создает предпосылки получения документов путем реструктуризации и редактирования из некоторой внутренней, машинной структуры в различные виды внешних структур, удовлетворяющих тем или иным требованиям.

Целесообразно, чтобы номенклатура документов была невелика, пересечение их по содержащейся информации было недвусмысленным и конструктивным.

Машинное документирование существенно упрощает технологию передачи готовых программных изделий потребителям. Эта технология должна воплощаться в стандартной процедуре генерации пакета программы на магнитную ленту потребителя в самовосстанавливаемом виде, т.е. в виде, который обеспечивает восстановление пакета программы на ЭВМ потребителя без особых усилий с его стороны.

Перечислим необходимые инструментальные средства, обеспечивающие языковую и программную поддержку технологического процесса создания пакета прикладных программ в системе ТЕКОМ.

Средства языковой поддержки включают язык разработки программы, языки программирования модулей (рабочие языки), язык конструирования сложных программ, язык построения лингвистического обеспечения при пакетировании программ, язык описания данных.

Язык разработки программы – это близкий к естественному язык типа PDL [9], использующий идеи структурированного программирования. Язык служит для четкого описания технических проектов программных изделий и является средством коммуникации для их разработчиков. В качестве рабочих языков программирования элементарных модулей в разрабатываемой версии ТЕКОМа принимаются ФОРТРАН и АССЕМБЛЕР. Язык конструирования сложных программ – это структурированный язык программирования высокого уровня, служащий для сборки модулей второго уровня. К этому языку предъявляется ряд специальных требований: динамическое распределение памяти для массивов с переменными границами, развитые средства диалога, организация контрольных точек и т.п.

Для построения лингвистического обеспечения – языка общения пользователей с программным фондом – удобным инструментом является язык R/TRAN и комплекс РТК ЕС ЭВМ [7]. Язык описания данных служит для написания дескрипторов, описывающих структуру обрабатываем-

мых информационных объектов и используемых программами обслуживания данных. Как уже отмечалось, программная поддержка производства программных изделий осуществляется мониторной системой, состоящей из описанного выше монитора и сопряженных с ним следующих системных комплексов программ:

- обслуживающих программы, осуществляющих ведение системных библиотек, ведение обрабатываемых данных, форманный вывод данных, обслуживание томов запоминающих устройств;

- обрабатывающих программы, осуществляющих трансляцию, редактирование связей, загрузку, отладку и т.п.;

- информатора - комплекса программ, осуществляющего автоматизацию документирования и информационно-справочную службу о программных изделиях в процессе их производства и эксплуатации.

Мониторная система является ядром создаваемых пакетов программ. Изготовление пакета заключается в функциональном расширении этого готового ядра проблемными программами из выбранной предметной области, а язык общения мониторной системы расширяется функционально нагруженным языком общения с пакетом прикладных программ.

Л и т е р а т у р а

1. ГЛУШКОВ В.М., ВЕЛЬБИНСКИЙ И.В. Технология программирования и проблемы ее автоматизации. - "Управляющие системы и машины", 1976, 6, с.75-91.

2. ЕРШОВ А.П. Технология разработки систем программирования. - 8 кн.: Системное и теоретическое программирование, Новосибирск, 1972, с. 136-184.

3. КАРПОВ В.Я., КОРЯТИН Д.А., САМАРСКИЙ А.А. Принципы разработки пакетов прикладных программ для задач математической физики. - "Ури. вычислит. математики и мат.физики", 1978, № 2, с.458-467.

4. СУХОМИН В.А. Метасистема для построения проблемно-ориентированных языковых систем. - "Программирование", 1976, 2, с.63-70.

5. ТАММ Б.Г., ТЫГУ З.Х. Пакеты программ. - "Изв.АН СССР.Техн. кибернетика", 1977, № 5, с.III-I24.

6. ЯНЕНКО Н.Н. Тенденция развития современной математики. - В кн.: Методологические проблемы научного познания. Новосибирск, "Наука", 1977, с.64-72.

7. РТК ЕС ЭВМ-технологический комплекс программиста. Киев. 1977 (ИК АН УССР).

8. Математическое обеспечение ЭВМ (тематическая подборка по зарубежным источникам). Калинин, "Центрпрограммистов", 1978.

9. Structured Programming Series. Vol.V-VII. IBM Federal Systems Division.

Ю. ФУКСМАН А.Л. Расслоенное программирование. -В кн.: Системное и теоретическое программирование, Новосибирск, 1974, с.16-31 (ВЦ СО АН СССР).

II. ВЛАДОВСКИЙ И.М. АСУ предприятием на базе ЕС ЭВМ. М., "Энергия", 1977.

Поступила в ред.-изд.отд.

24 октября 1978 года