

где $K_{ЭВМ}$ — полная стоимость ЭВМ.

В целом, использование единого комплекса вычислительных средств для автоматизации нескольких научно-исследовательских разработок может обеспечить наиболее полную и равномерную загрузку основных фондов, повысить надежность работы комплекса и дать наибольший эффект.

УДК 519.6:681.142.4

В.А.Виттих, В.В.Куликов

О ПОДГОТОВКЕ СПЕЦИАЛИСТОВ В ОБЛАСТИ ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ

(г.Куйбышев)

Технология программирования — стремительно развивающаяся область инженерной деятельности, направленная на улучшение качества программного обеспечения современных вычислительных комплексов и повышение производительности процессов его разработки. Технология программирования охватывает такие сферы деятельности, как проектирование, разработка и документирование систем программного обеспечения (СПО), что приводит к потребности развития в рамках новой дисциплины методов решения, характерных для традиционных технических дисциплин.

Исследования, проведенные в нашей стране [1] и несколько ранее за рубежом [2], показывают, что термин "технология программирования" (принятый в СССР) или "программотехника" (принятый в США) включает в себя обширный перечень проблем по следующим направлениям:

системный анализ проблемы, описание функций проектируемой системы, расслоение проблемы на относительно самостоятельные части, выработка приемлемых решений;

проектирование программного обеспечения на основе строгих (формализованных) методов для построения логической структуры СПО, создание и оценка вариантов СПО;

использование (а при необходимости и разработка) программной методологии для надежной разработки корректных программ, включая поэтапное улучшение и структурное программирование;

тестирование и верификация программ;

автоматизация процесса создания комплексного программного обеспечения (включая процессы контроля, редактирования, отладки, средства общения с СПО);

оценка качества и эффективности функционирования СПО;

средства и способы документирования СПО;

управление разработкой программного обеспечения.

Этот далеко не полный перечень задач, образующих понятие технологии программирования как инженерной дисциплины, позволил специалистам в области создания больших программных систем сделать следующий вывод [1] : "...Задача программной техники (технологии программирования (комментарий наш), заключается в реализации общесистемного подхода к построению программного обеспечения, при котором оптимизация всех ресурсов проводится как на стадии разработок, так и на стадии эксплуатации. Несмотря на то, что программная техника пока остается скорее неосуществленной мечтой, чем реальностью, концепция инженерного подхода к созданию программного обеспечения резко изменила точку зрения многих людей на эту проблему. Постепенно методы программной техники находят применение во многих значительных проектах, заменяя собой анархическую практику прошлого. Все больше и больше программная техника рассматривается как комплексная техническая деятельность, способствующая развитию согласованной методики создания программного обеспечения и присущая всем стадиям цикла жизни программного обеспечения - от составления технического задания до эксплуатации системы..."

Истоки и процесс становления технологии программирования являются сейчас предметом изучения специалистов. Но следует отметить, что виртуозное владение языками программирования высокого уровня (или, как было принято их называть

в начале 60-х годов, "универсальными языками программирования") и математическими методами обработки данных не являются достаточными условиями для существенного сокращения сроков реализации проектов программных систем. Технология программирования должна обеспечить в конечном итоге более высокую степень автоматизации процесса создания программных систем при одновременном улучшении технико-экономических показателей программных систем.

Внедрение в практику новой инженерной дисциплины потребовало пересмотра представлений о роли программирования в различных сферах производственной деятельности. Анализ истории развития программирования [1, 3] определил ряд тенденций:

к использованию результатов, полученных с помощью ЭВМ, прибавилось в промышленно развитых странах до 30% от общей численности работающих. При этом почти половине из них не требуется иметь профессиональных навыков работы на вычислительной машине, но необходимы знания о возможностях ее использования;

при разработке больших программных систем профессиональным программистам требуются фундаментальные знания по теории проектирования сложных систем, а не набор сведений о программистских "трюках", сводящих программирование к своеобразному искусству;

в последнее время отмечается существенное повышение затрат на разработку программного обеспечения в сравнении с затратами на разработку аппаратных средств вычислительных комплексов. Так, по оценкам ИК АН СССР, если затраты на производство программного обеспечения в 60-х годах составляли лишь около 20% от общих затрат на разработку вычислительных комплексов, то в 70-х годах затраты на программное обеспечение возросли до 60% и продолжают стремительно расти;

индивидуальный характер труда программистов приводит к функциональному дублированию около 60-70% разработок, при этом на последующее сопровождение в период эксплуатации таких программных систем затрачивается свыше 70% средств от общей стоимости работ, выполняемых на вычислительном комплексе;

за последние годы штаты системных программистов и технологическая оснащенность групп существенно возросли, что, однако, не привело к заметному скачку в развитии программирования.

По данным фирмы IBM индивидуальное мастерство "звезд" программирования, выдвинутых на руководящие должности, не приводит к

адаптации индивидуального опыта руководителя к условиям плановой коллективной работы группы программистов.

Разработчики крупных программных проектов типа *US-360* (США), Альфа-системы программирования (СССР), *R* - технологического комплекса программирования (СССР) признают, что узловыми проблемами являются:

повышение производительности труда программистов;

повышение надежности сроков реализации программных проектов.

К ним следует добавить также качество подготовки системных программистов.

Типичная схема обучения программиста, принятая как в зарубежной, так и в отечественной практике, строится на основе тщательного изучения систем счисления, изучения приемов программирования в машинно-ориентированном языке типа Ассемблер и одного, реже двух языков высокого уровня. Крупнейший разработчик ряда транслирующих систем Н. Вирт, создавший в последнее время язык системного программирования ПАСКАЛЬ [4] отмечает: "Студентов нужно обучать конструированию алгоритмов методично и систематически, знакомя их с задачами и приемами, типичными для программирования и не зависящими от конкретных приложений... Достижение совершенства в использовании специфических особенностей какого-либо конкретного языка не должно стать основной целью программирования..."

Одной из попыток решения перечисленных задач в совокупности явилась разработка плана по подготовке программистов [2] :

Общий курс

1. Введение в программную технику.
2. Творческий подход к решению задач.

Прикладная вычислительная техника

1. Архитектура вычислительных систем.
2. Применение языков программирования.
3. Разработка операционных систем.
4. Организация файлов и баз данных⁺
5. Системы связи⁺

⁺ Знаком (+) отмечены факультативные курсы.

Методы управления

1. Управление разработкой программного обеспечения.
2. Составление технологических описаний и программной документации.
3. Экономические аспекты программотехники.
4. Принципы теории управления.

Аналитические методы

1. Надежность программного обеспечения.
2. Прикладные методы формального анализа.
3. Анализ и оценка функционирования систем.

Разработка программного обеспечения

1. Требования к программному обеспечению и его характеристики.
2. Методы разработки программного обеспечения.
3. Методология программирования.
4. Подтверждение корректности программного обеспечения и его проверка.
5. Лабораторный курс (практикум).

Для подготовки специалистов, ориентированных на разработку СПО современных систем реального времени, приведенную программу целесообразно дополнить разделами по функциональным пространствам, методам представления информационных структур, расширить лабораторный курс методами диалогового проектирования комбинационных схем и магазинных преобразователей. Подготовка таких специалистов может осуществляться на базе специальностей 0646, 0647, 0608.

Концептуальная основа технологии программирования

Основной целью любой из технологий является создание программного продукта, которому свойственно следующее:

единая концепция, основанная на использовании формального математического аппарата для всего технологического цикла изготовления СПО: проектирования, кодирования, отладки, документирования, эксплуатации и адаптации к изменяющимся условиям вычислительной среды;

пошаговая (послойная) разработка СПО, как правило, сверху вниз, начинающаяся с приближенной модели решаемой задачи с последующим пошаговым уточнением исходной модели до получения требуемой точности. При этом независимо от степени детализации ("слоя") проектируемой СПО используется единая методология решения задачи;

степень формализации задач такова, что появляется возможность автоматизации проверки правильности сборки программного продукта и его частичного или полного тестирования;

пошаговая разработка СПО позволяет совместить процессы отладки частичного программного продукта с его детализацией и тем самым уменьшить объем отладок до реально приемлемых. Это свойство принципиально отличает технологию программирования от традиционного программирования и позволяет существенно сократить сроки реализации программных систем;

появляется возможность устранить функциональное дублирование программ, так как ранее разработанные и отлаженные программные компоненты любого уровня детализации включаются во вновь разрабатываемую систему автоматически.

Работа I Всесоюзной конференции по технологии программирования, освоение R - технологического комплекса программиста, изучение базы данных ИНЭС, являющейся предшественницей R - технологии, выявили эффективность нового направления в программировании и необходимость быстрее внедрения подобных технологий в практику программирования.

Открытые проблемы технологии программирования

В настоящее время библиография работ по технологии программирования и смежным вопросам насчитывает более 500 публикаций [1, 6]. В различных организациях страны проводятся работы по совершенствованию методов проектирования программных систем.

В процессе исследований, проведенных в нашей стране и за рубежом [1, 6, 7] было установлено, что эффективность технологии программирования и возможность автоматизации процессов создания программных систем в инструментальном комплексе обеспечивается свойствами формальной алгебраической системы. Условия "технологичности" создаваемой программной системы и ее основных компонент, сформулированные в первых работах по структурному программирова-

нию [1, 5, 6], привели к появлению целого ряда методологий и технологий построения программных систем, а главное — стимулировали развитие работ в области формальных систем (исчислений, алгебр, математических структур и т.п.), ориентированных на решение наиболее актуальных задач программирования.

Новые технологии позволяют успешно решать задачи по разработке транслирующих систем, систем организации диалога пользователя — непрограммиста с вычислительным комплексом, систем подготовки данных и т.п. Разрабатывается формальная основа и инструментальные системы проектирования программно-технических систем типа ИВК (на базе СМ ЭВМ) с аппаратурой КАМАК. В процессе работ, проведенных в различных организациях страны [6], проверялись достоинства и недостатки технологии программирования в существующем варианте. При этом выявлена недостаточная теоретическая проработка существующих технологий в целях реализации систем реального времени, представляющих собой программно-технические системы с мультипроцессорным режимом обработки данных.

Суть проблемы заключается в необходимости программирования параллельных взаимодействующих процессов, характерных для функционирования дискретных программно-электронных систем. Другая проблема состоит в том, что структурное программирование абстрагируется от конкретных приложений [4], т.е. сводится по существу к формальной языковой системе, учитывающей лишь синтаксический аспект информационных структур. Исследования, проведенные в последние годы отечественными [6, 8] и зарубежными учеными [7, 9], показали, что в системах реального времени и транслирующих системах важно учитывать и задавать семантические связи между элементами синтаксических конструкций и выполнять **структурирование** (композицию и декомпозицию систем) как по синтаксическим, так и по семантическим критериям. Подобное решение проблемы ведет к тому, что структурное программирование, являясь удобной (технологичной) формой конструирования СПО, должно быть дополнено соответствующим семантическим содержанием, тогда задача структурного программирования сводится к целенаправленному преобразованию исходной алгебраической системы, полученной по исходному описанию задачи, в одну или несколько других алгебраических систем типа булевых алгебр и более просто интерпретируемых заданной вычислительной средой. Процессу такого преобразования присущи структурные свой-

ства, он поддается автоматизации. Метод прошел промышленную проверку при создании программно-технических автоматов управления технологическими процессами в химическом производстве, на транспорте. Он особенно пригоден для проектирования дискретных технических систем на основе современной микроинтегральной технологии.

В работах [7, 8] рассмотрены типичные для специалистов по обработке данных математические модели дискретных динамических систем, а методы их преобразования в алгоритмические и далее в языковые модели (или языки программирования) показывают возможность и необходимость учета семантических свойств объектов математических моделей (структур данных и алгоритмов). Следовательно, подобные модели не требуют изучения новых дисциплин, требуется лишь учитывать правила их преобразования с учетом ограничений, характерных для применяемой технологии.

Краткий обзор методов реализации программных проектов

Анализ свойств технологии программирования проводился на основе наиболее известных методов разработки хорошо зарекомендовавших себя в зарубежной и отечественной практике проектирования больших программных систем. Суть каждого из них в следующем:

метод "х и р у р г и ч е с к о й б р и г а д ы" или метод главного программиста [3], когда один программист пишет, отлаживает, редактирует и документирует программу на **структурированном** языке типа ПД/Г. Остальные - его дублер (ассистент), администратор, редактор, секретари, архивариус, инструментальщик и т.д., всего 18-20 человек, выполняют вспомогательную работу. Метод дал положительный результат при реализации базы данных газеты "Нью-Йорк Таймс". Недостаток - ограниченная производительность главного программиста. Достоинства - единая концептуальная основа всего проекта;

н и с х о д я щ е е п р о е к т и р о в а н и е представляется как последовательность шагов, уточняющих проект. Метод **предложен** и реализован Н.Виртом в языке ПАСКАЛЬ. Язык ПАСКАЛЬ входит в состав математического обеспечения многих современных вычислительных комплексов, в том числе и в математическое обеспечение ЭВМ БЭСМ-6. Достоинства метода - концентрация внимания проектировщика на сравнительно небольшом участке ("слое") задачи;

венский метод или метод определения языков программирования интерпретирующими автоматами [9]. Метод разработан и дал положительный результат при реализации самого мощного из активно используемых языков программирования высокого уровня ПЛ/Г. Недостатком можно считать отсутствие возможности введения программистами собственных синтаксических конструкций, т.е. ограниченная проблемная ориентация. Достоинство ПЛ/Г - диапазон применений от численных расчетов до реализации транслирующих систем;

язык рекурсивных функций РЕФАЛ - компонента базы данных ИНЭС [10] - является предшественником R - технологии в СССР. Достоинства РЕФАЛа - рекурсивное описание алгоритмов и данных практически произвольной глубины вложенности, возможность самопреобразования тестов алгоритмов, описание взаимодействующих параллельных процессов. Используется для реализации инструментальных и транслирующих систем [6]. Недостаток - слабая ориентация на оформление документации разработок;

R - технология [11] - наиболее приспособленная для изучения система производства программ, ориентированная на обработку данных, символьные преобразования. Достоинства - простота изучения, освоения. Обеспечивает легкое представление структур данных, возникающих в процессе проектирования. Недостаток R - технологии, как и всех вышерассмотренных систем - в отсутствии развитых средств параллельного программирования, средств проверки согласования структур данных и алгоритмов;

проект Shuttle [7] - система проектирования и отладки математического обеспечения космического комплекса *Space Shuttle*. Дальнейшее развитие технологии программирования, получившее название *HOS* - методологии (*Higher Order Software*). Реализованы алгоритмы автоматической проверки согласования структур в статическом и динамическом режимах, алгоритмы автоматизированной локализации ошибок. Существенным отличием от известных технологий является полная формализация процессов проектирования и режим параллельных вычислений в темпе реального времени. Получено снижение числа ошибок в отладке на 70% в сравнении со структурным программированием, использовавшимся для проектирования аналогичной системы (проект *Aravello*) ранее.

Отсутствие средств описания семантических преобразований, средств описания параллельных процессов, средств проверки согласо-

вания структур данных в существующих технологических системах ограничивает их полное использование для проектирования СПО ИВК, функционирующих в темпе реального времени. Фактически, теоретические основы проектирования современных программных систем изложены кратко лишь в литературе по R -технологии. Вследствие этого **затрудняется** изучение, развитие и эффективное использование нового направления. Например, попытки построить транслирующую систему с использованием R -технологии осложняются тем, что все вопросы преобразования синтаксического графа в структуру ЭВМ решаются каждым программистом самостоятельно и, как показывает целый ряд работ, весьма неквалифицированно и бессистемно.

В ы в о д ы

Анализ основных свойств современных методологий и технологий позволил установить наиболее существенные пробелы в подготовке специалистов по программированию, обусловившие возникновение своеобразного "кризиса" программирования [2]. К числу основных факторов, отрицательно влияющих на качество подготовки специалистов в области программирования, можно отнести следующие:

разрыв между методами анализа и синтеза систем преобразования информации и методами их программной реализации;

отсутствие "сквозной" технологии проектирования программно-аппаратных вычислительных систем и, в первую очередь, систем реального времени;

отсутствие методической литературы, пособий и т.п. по проектированию программно-аппаратных систем на основе единой методологии;

нехватку квалифицированных преподавателей, имеющих достаточный практический опыт.

Отметим, что необходимый теоретический и практический материал, составляющий основу современных технологий, существует в "рассеянном" по сборникам и журналам виде, и его систематизация, а также разработка учебных планов и академических курсов по программной технике – единственно возможное решение на пути преодоления "кризиса" и выработки приемлемой инженерной методики проектирования больших программно-аппаратных систем.

Л и т е р а т у р а

1. Глушков В.М. Фундаментальные исследования и технология программирования: Тезисы докладов I Всесоюзной конференции "Технология программирования".-Киев:Изд-во ИК АН СССР, 1979.
2. Вассерман А.И., Фримэн П. Подготовка специалистов в области программотехники: состояние и перспективы.-ТИИЭР, № 8, том.66, 1978.
3. Брукс Ф.П. Как проектируются и создаются программные комплексы /Пер. с англ.-М.:Наука, 1979.
4. Вирт Н. Систематическое программирование. Введение /Пер. с англ.-М.:Мир, 1977.
5. Дал У., Дейкстра Э., Хоор К. Структурное программирование/Пер.с англ.-М.:Мир, 1975.
6. Автоматизация производства пакетов прикладных программ (автоматизация производства трансляторов):Тезисы докладов.Таллин: Таллинский политехнический институт, 1980.
7. Шевченко А.М. Математическое обеспечение космического комплекса *Space Shuttle*: Тематический сборник ИППИ.-М., 1978.
8. Горбатов В.А. Семантическая теория проектирования автоматов.-М.:Энергия, 1979.
9. Оллонгрэн А. Определение языков программирования интерпретирующими автоматами /Пер. с англ.-М.:Мир, 1978.
10. Базисный РЕФАЛ. Описание языка и основные приемы программирования. (Методические рекомендации).-М.:ЦНИИАС, Вып.5, т.33, 1974.
11. Вельбицкий И.В., Ходаковский В.Н., Молмов Л.И. Технологический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6.-М.:Статистика, 1980.
12. Мановцев А.П. Основы теории радиотелеметрии. (Представление и обработка сообщений, информационное обслуживание).-М.: Энергия, 1973.