

ОПЫТ ИНТЕГРАЛЬНОГО ПОДХОДА К АКТУАЛЬНОЙ ПРОБЛЕМАТИКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ВВЕДЕНИЕ

Уже давно стало ясно, что переход к новому поколению вычислительной техники (см. таблицу), хотя и обусловлен сменой элементной базы и возрастанием на порядок предельных характеристик высокопроизводительных вычислительных машин, все же одним этим далеко не ограничивается. Появление нового поколения ЭВМ влечет значительные перемены в способах применения вычислительной техники, качественные изменения в номенклатуре и организации программного обеспечения.

Дополнительным обстоятельством последних десяти лет стала консервативная роль программного обеспечения. Интеллект и знания, воплощенные в программах, становятся самой ценной компонентой вычислительного дела, залогом преемственности и стабильности в применении вычислительной техники.

Одним из последствий этого обстоятельства стало увеличение временной дистанции между сменами поколений и более постепенный переход от одного поколения к другому, другим – учет достаточно далекой перспективы и охват всех аспектов программирования.

И наконец, программно-целевой подход, применяемый сейчас в планировании, требует формирования проблематики не столько путем экстраполяции достигнутого, сколько на основе сформулированных на какой-то период конечных целей развития с последующей их интерполяцией на промежуточные отрезки времени и подцели.

Работа, предлагаемая вниманию читателей, представляет попытку дать интегральное представление о проблематике программного обеспечения, определяемой потребностями развития на десяти-пятнадцатилетнюю перспективу.

Необходимость совместить объем журнальной статьи с широким охватом тем неизбежно привела к конспективной манере изложения. Автор надеется, однако, что это не помешает читателю воспринять материал статьи как канву, по которой каждый специалист сможет исполнить свой узор, не порывая с представлением об общей картине.

Ряд положений статьи возник в результате бесед со многими специалистами. Особенно автор хотел бы отметить полезные замечания и суждения, сделанные В. И. Дракиным, В. В. Липаевым, Л. Д. Райковым и М. Р. Шура-Бурой. Придавая материалу в целом обзорный характер, автор не претендует на приоритет ни по какому конкретному утверждению, сохраняя, однако, за собой всю ответственность за его правильность.

ОБЩИЕ ПРОБЛЕМЫ РАЗВИТИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Постоянно действующей задачей развитого программного обеспечения является достижение заданного уровня выпуска программных средств в полной номенклатуре и объемах, определяемых совокупностью производимых вычислительных средств и областей их применения.

Эта главная задача требует решения следующих общих проблем.

1. Организация основного объема профессионального программирования как промышленного производства продукта производственно-технического и потребительского назначения

Промышленный характер разработки и производства программного обеспечения вызван потребностью многократного увеличения объема производства с учетом его возрастающей сложности и необходимостью гарантированного соблюдения его качества и работоспособности в течение длительного времени при условии высокой тиражности (средний тираж порядка 10 тыс. экземпляров) и полного отчуждения применяемого программного продукта от его изготовителей. Строгая регламентация промышленного производства программного продукта не должна исключать существование и особых форм разработки экспериментальных или малотиражных программ небольшой степени сложности.

Характеристика	Поколения (годы)				
	Пятое (1991 – 2005)	Четвертое (1978 – 1990)	Третье (1965 – 1977)	Второе (1956 – 1964)	Первое (1950 – 1955)
Максимальная скорость (опер./с)	1000 млн.	100 млн.	10 млн.	1 млн.	10 тыс.
Максимальный объем ОЗУ (байт)	1000 М	100 М	10 М	0,5 М	50 К
Максимальный объем ВЗУ (байт)	1000 Г	100 Г	10 Г	1 Г	10 М
Информационная вооруженность (опер./с·чел.)	1 млн	10 тыс.	50	1/2	1/100
(объем ПО / чел., байт)	1 М	100 К	10 К	100	0
(объем информации /чел., байт)	10 М	1 М	100 К	5 К	1 К
(входов с СПД/чел.)	1,5	0,7	1/10	1/100	1/1000
Индивидуальная производительность (байт програм../год)	50 тыс	20 тыс.	6 тыс.	3 тыс.	2 тыс.
Средний тираж программ	50 тыс.	10 тыс.	1 тыс.	100	10
Оборудование	СБИС, в том числе программируемые, память 256 К байт/чип, доменная, оптическая, суперЭВМ, речевая связь, вывод полиграфического качества, машинное зрение	БИС, память 64 К байт/чип, микропроцессоры, мега-мини-ЭВМ, ПЭВМ, лазерные принтеры, дистанционная отладка, диски 1 Г байт, винчестерские	ИС, полупроводниковая память 16К байт/чип, мини-ЭВМ, флоппи-диски, дисплеи	Транзисторы, диски	Электронные лампы, ферритовая память, барабаны, ленты
Архитектура	Гетерогенная, потоковая, рекурсивная. Сопряжение с автоматической системой связи и средствами массовой информации.	Конвейерная, векторная, многопроцессорная. Распределенная обработка, локальные сети.	Микропрограммная. Ряды совместных ЭВМ, системы разделения времени, сети ЭВМ	Мультипрограммная. Операционные системы.	Дж. Фон Неймана

Программное обеспечение	Универсальные языки спецификаций, базы знаний, системы доказательного программирования, обработка изображений, интеллектуальные пакеты (экспертные системы, автоматический выбор методы и синтез программ, интерфейс на естественном языке).	Переносимое ПО на базе унификации операционных систем, баз данных и языков программирования, организационная и инструментальная поддержка программирования, языки спецификаций для классов задач, обработка текста, растровая машинная графика.	Базы данных, ППП, машинная графика, диалоговые системы, программный продукт, макроассемблеры, унифицированные языки программирования, переносимость на уровне ЭВМ.	Языки программирования и трансляторы, программное обеспечение.	Автокоды, библиотеки программ.
Применение	Начальная информатизация общества, ГАП, интеллектуальные роботы, АРМ в промышленности, образовании и медицине	Встраивание в машины, Глобальные системы АРМ в делопроизводстве, в научных исследованиях, электронике, промышленные роботы.	АСУТП, информационные системы, АРМ проектирования. Автоматизация эксперимента.	АСУ, инженерные и экономические расчеты	Наука

2. Создание научно-методологических основ практического программирования и их внедрение в систему образования в систему производства программного продукта

В настоящее время при разработке программ преобладает интуитивный подход, а в оценке качества и пригодности программного продукта наибольшее влияние оказывают вкусы и предпочтения разработчика. Это положение вещей сложилось тогда, когда программы составлялись главным образом для себя или для сравнительно узкого круга потребителей, непосредственно связанных с разработчиком. Повышение общезначимости программного продукта и степени зависимости отдельных потребителей и народного хозяйства в целом от программного обеспечения требует изменения сложившегося положения, поскольку в противном случае стоимость разработки, издержки сопровождения и ущерб от отказа вычислительных программных средств в период эксплуатации возрастут катастрофически.

Более конкретно, создание научно-методологических основ программирования должно достигать следующих целей.

Доказательность. Разработчик должен уметь представить систему рассуждений и, если необходимо, экспериментальных данных, гарантирующих правильность функционирования программы в пределах заданных степени риска, всеобщности и тиражности программных средств.

Систематичность. Способ составления программ должен быть систематичным, т. е. способным к повторению, доступным для контроля и учета внешних факторов, устойчивым во времени, доступным для изучения и освоения средствами образования.

Познаваемость. Программы должны быть познаваемы, допуская на практике передачу ответственности за программный продукт другому лицу или другой организации.

Общезначимость. Программы должны быть общезначимы, допуская на практике автоматическое и правильное функционирование во всем диапазоне условий применения с учетом их высокой тиражности.

Многократность. Компоненты программного обеспечения (модули и процедуры) должны производиться с учетом их многократного последующего использования в качестве полупродукта при создании программного обеспечения. Необходимо стремиться к тому, чтобы в среднем не менее двух третей содержимого оригинала дистрибутивной ленты составляли модули повторного использования, включая автоматически адаптированные.

Эргономичность. Неотъемлемой функцией программы помимо продуктивной автоматической работы управляемого устройства является обеспечение взаимодействия устройства с человеком-оператором или пользователем. Учитывая, что "площадь соприкосновения" программного продукта с пользователями в перспективе возрастет минимум на два порядка (от десятков тысяч до миллионов), необходим глубокий и опережающий анализ всех человеческих факторов, определяющих эффективность взаимодействия человека с ЭВМ или с машинами, ими управляемыми.

3. Определение номенклатуры и объема производства программного обеспечения в зависимости от совокупности производимых вычислительных средств и областей их применения

При учете производства программных средств, а также при оценке тенденций развития следует учитывать долю покупного и накопленного программного обеспечения в создаваемых программных средствах. Под покупным подразумевается изготовленное независимо программное обеспечение, использование которого в конечном продукте требует адаптации или освоения. Накопленное программное обеспечение – это запас модулей, процедур и других ранее отработанных компонент, которые включаются в конечный продукт целиком или средствами автоматической адаптации, не утрачивая при этом своей правильности.

В целом следует стремиться к сокращению доли адаптируемого и повышению доли накапливаемого продукта, поскольку первый характеризует степень зависимости, а второй обеспечивает чистую экономию труда. Однако в перспективе путем установления международных стандартов на переносимость, надежность и познаваемость программного продукта необходимо содействовать возможности сближения продуктивной полезности покупного и программного продукта. Естественно,

что такая система международной кооперации в первую очередь должна сложиться между социалистическими странами.

В качестве общей цели следует потребовать, чтобы доля накапливаемого и приравненного к нему продукта составляла бы в перспективе порядка двух третей годового производства программных средств.

Номенклатуру программных средств целесообразно подчинить следующей общей классификации: операционные системы, системные компоненты, средства программирования, пакеты прикладных программ общего назначения, проблемно-ориентированные пакеты прикладных программ.

Операционные системы образуют ядро системного программного обеспечения вычислительного средства и состоят из собственно операционной системы ЭВМ, а также библиотек обслуживающих и проверочных программ. Организационно-технической особенностью операционных систем является то, что они разрабатываются совместно с аппаратурой и входят в обязательный комплект поставки вычислительного средства.

Системные компоненты отличаются от операционной системы меньшей степенью интегрированности и способностью поставляться как отдельное программное средство. В то же время от ППП их отличает близость и тесная связь с операционной системой. Развернутая номенклатура функциональных компонент должна содержать такие средства, как теледоступ и работа в сети, машинная графика, обработка текстов, а также системы управления базами данных, а в перспективе – средства обработки факсимильной информации, обработки изображений, речевой и звуковой связи и системы управления базами знаний.

Средства программирования представляют собой комплекс разнообразных средств и отличаются от других ППП своей актуальностью, общезначимостью, а также необходимостью быть предметом ответственности ведомства-изготовителя вычислительных средств. В дополнение к сложившимся средствам программирования в виде алгоритмических языков, трансляторов, систем отладки, редакторов программных текстов должны разрабатываться или получать более широкое распространение интегрированные системы разработки программ (ИСРП), или так называемые программные обстановки.

ИСРП в развернутом составе – это организованная совокупность функциональных и языковых средств, обеспеченных аппаратной и информационной поддержкой, реализующих законченный технологический цикл проектирования, разработки и сопровождения программного средства в рамках единой методологии.

ППП общего назначения выделяются из других прикладных программных средств своей общезначимостью и массовостью, делающими целесообразным концентрацию выпуска этих пакетов в ведомстве-изготовителе вычислительных средств на основе широкой кооперации, главным образом с академиями наук и минвузами.

Следуя В. И. Дракину, объединяем системные компоненты, средства программирования и ППП общего назначения в базовые программные системы, создание или выпуск которых должны стать уделом специализированных коллективов профессиональных программистов.

Проблемно-ориентированные ППП обеспечивают решение устойчивых классов задач, характерных для той или иной отрасли народного хозяйства. Их разработка осуществляется главным образом предприятиями-изготовителями программных средств, принадлежащими отрасли-потребителю вычислительной техники.

К настоящему времени разработка вычислительных и программных средств расчленена между рядом ведомств и ведущих коллективов. Это расчленение частично отражает классификацию вычислительных средств по классам или даже конкретным моделям, а частично сложилось исторически. С учетом перспективы целесообразно поддерживать следующую классификацию:

- суперЭВМ;
- единый ряд многомашинных вычислительных комплексов;
- единая система ЭВМ;
- система малых ЭВМ;
- микропроцессорные серии (микросерии);
- персональные ЭВМ;

– вычислительные машины других архитектур.

Признавая целесообразным вести по каждому типу ЭВМ свою программу развития и производства, в том числе и программных средств, необходимо закладывать в технические решения спектр возможностей, допускающих совместное применение вычислительных средств разных классов и обеспечивающих в разумном объеме переносимость и повторную используемость программных средств. В частности, такие возможности должны предусматривать следующее:

– стандарты на сопряжение и на объединение в сеть ЭВМ разных моделей;

– унификацию внешних запоминающих устройств и периферийного оборудования;

– стандарты на языки программирования, обеспечивающие переносимость программных модулей и процедур;

– унификацию систем управления базами данных, обеспечивающую переносимость информационных фондов и программ, использующих такие фонды;

– унификацию операционных систем, обеспечивающую переносимость ППП и диалоговых систем;

– развитие микропрограммной реализации машинной архитектуры, облегчающее создание гибридных систем и выполнение программ на основе эмуляции.

4. Рост производительности труда программистов, значительное повышение его надежности при умеренном росте числа профессиональных программистов и при значительном сокращении удельной стоимости сопровождения и эксплуатации

Для указанных показателей характерна глубокая взаимность, каждый показатель должен точно учитываться и тщательно прогнозироваться. Взяты вместе они характеризуют технический уровень и общественную эффективность программирования. Отправной точкой в достижении указанной цели должна стать разработка методики определения этих показателей.

Средняя производительность труда программистов при разработке программного продукта исчисляется как отношение объема законченного продукта, выраженного в килобайтах, к совокупным трудозатратам программистов, выраженным в человеко-годах и охватывающим весь период их участия в разработке.

Надежность программного средства на какой-то период времени его эксплуатации можно исчислять как величину, обратно пропорциональную числу отказов от выполнения (зацикливания или нештатные остановки), случившихся за взятый период и отнесенных к количеству запусков программы или другому показателю, характеризующему интенсивность использования средства.

Под понятием "профессиональный программист" подразумевается специалист, по главному роду своей деятельности связанный с созданием отчуждаемого программного продукта. Со временем профессиональный статус программиста должен подтверждаться образовательным или аттестационным сертификатом.

Под затратами на сопровождение и эксплуатацию подразумеваются все совокупные общественные затраты как со стороны изготовителя, так и пользователя, вызванные применением программного продукта и поддержанием его в работоспособном состоянии. Удельные затраты исчисляются как общие затраты, отнесенные к единице объема программного продукта (например, байт программного текста) с учетом его тиражности. По мере развития методики оценки расходы на эксплуатацию могут быть отделены от расходов на сопровождение.

Наиболее трудно достижимыми показателями и в то же время определяющими остальные являются производительность программистов и надежность программного продукта. Следует отметить их разнонаправленность: обеспечение высокой надежности тормозит рост производительности.

Основными факторами повышения производительности считаются: владение несколькими языками программирования, виртуализация ресурсов ЭВМ, инструментальная поддержка, степень доступности и реактивности инструментальной ЭВМ, следование организационной и методологической дисциплине, повторное использование компонент программного обеспечения, опыт, способности, условия труда

и меры стимулирования.

Из объективных факторов, тормозящих рост производительности, прежде всего следует указать на необходимость удовлетворять ограничениям на время срабатывания программы и на расход памяти, необходимость высокого показателя надежности, сложность программного изделия. Последний фактор дает возможность различать при более точных оценках производительности по крайней мере три уровня сложности программного продукта:

- программы разового применения среднего объема (сложность 1);
- программные комплексы большого объема постоянного применения (сложность 3);
- программные комплексы реального времени (сложность 6).

Сопоставляя с оценками специалистов и с учетом "фольклорной статистики" можно оценить среднюю производительность (сложность 2) в настоящее время в 10 К байтов в год. Учитывая сложившиеся тенденции роста производительности, а также большое количество недоиспользованных резервов, реально поставить задачу удвоения производительности программирования к началу 90-х годов и утроения – к 2000 году.

При сопоставлении расчетной производительности с оценкой потребного числа программистов следует иметь в виду, что указанная производительность достигается только в профессиональных коллективах, в которых концентрируется производство программного продукта. При переводе оценок производительности и валового программного продукта в число требуемых специалистов нужно учитывать утечку программистов в непроизводственную сферу, т.е. в области применения ЭВМ, требующие программирования разовых задач, в смежные области деятельности (освоение предметных областей, системный анализ, подготовка исходных условий и данных), в сферу образования, а также естественную утечку кадров.

В настоящее время отсутствует сложившаяся система определения надежности программ. Однако косвенные оценки, а также отдельные случаи уже показывают, что издержки, вызванные недостаточным качеством и ненадежностью программного обеспечения, исчисляются сотнями миллионов рублей в год. Поэтому, какова бы ни была методика исчисления надежности, необходимость ее резкого повышения по сравнению с существующим положением становится для программистов задачей номер один.

Приведем факторы повышения надежности – это тщательное изучение и, по возможности, формализация предметной области применения ЭВМ, заблаговременный системный анализ задачи применения ЭВМ, тщательный учет всех условий использования ЭВМ при решении задачи, формализация технических заданий и спецификаций программного продукта, применение методологии доказательного и систематического программирования, автоматизация процесса программирования, применение средств независимого контроля, использование ранее разработанного и достоверного программного продукта, интенсивный и длительный период комплексных испытаний программного продукта до его тиражирования.

Кризисное состояние программирования объясняется быстрым ростом затрат на сопровождение программного продукта. Этому способствуют недостаточная надежность, дублируемость программных разработок, рост тиражности программного продукта и его сложность, несоответствие условий применения проектным положениям, недостаточная познаваемость программ, необходимость изменчивости программного продукта в его жизненном цикле. Некоторые из факторов носят объективный характер, другие отражают период становления промышленности производства программ. Так, последнее обстоятельство требует различать типы программного продукта по степени его изменчивости и массовости. Здесь уместно использовать организационно-технические достижения зрелых отраслей промышленного производства, прежде всего машиностроения, для более быстрого и сознательного использования полезных и сложившихся методов организации производства. В частности, программные средства тоже следует делить на уникальные, единичные, серийные и массовые, при этом каждый вид требует своих подходов в сочетании изменчивости со стабильностью, своих подходов к организации сопровождения. Особо выделим производство уникальных и единичных изделий, для которых обеспечение роста и непрерывной изменчивости требует глубоких подготовительных мер еще

на стадии проектирования и организации сопровождения такого же интенсивного и творческого процесса, как и разработка. В то же время для серийных и особенно массовых изделий следует добиваться максимальной стабильности программного средства, его высокой надежности и тщательной проверки в период длительных и интенсивных испытаний.

Важным показателем достижения такой надежности и стабильности массового программного продукта является неуклонное снижение удельной стоимости его сопровождения, которое с учетом роста средней тиражности программ должно быть не менее чем десятикратным по сравнению с существующим положением.

5. Обеспечение массового применения вычислительных средств внешними пользователями без посредничества программистов

В настоящее время средние условия применения ЭВМ требуют от потребителей знаний профессиональных программистов, которые оказываются необходимыми либо для ориентировки в тонкостях эксплуатации штатного программного обеспечения, либо для нетривиального программирования в терминах используемого прикладного пакета – будь то проблемно-ориентируемый пакет или система программирования, либо для адаптации полученного программного обеспечения.

Это положение сложилось закономерно, когда, с одной стороны, применение машин носило главным образом пробный, поисковый и экспериментальный характер, а способы постоянного, стабильного и четко определенного применения ЭВМ еще не сложились. С другой стороны, и производство программных средств носило открытый, незаконченный характер, не обладая способностью предугадать всю совокупность условий применения программного средства.

Сейчас положение уже меняется как по складывающейся невозможности приставить программиста к каждой ЭВМ, так и по наступающей возможности обеспечить "беспрограммируемое" применение ЭВМ во многих областях массовой деятельности на основе научно-технического прогресса.

Изменение положения дел подтверждается американскими данными (SETC Newsletter, 1983, 7, № 4, p. 4), согласно которым отношение между числом ЭВМ и числом программистов менялось следующим образом: 1955 г. – 1:10, 1980 г. – 3:1 (940 тыс. ЭВМ и 240 тыс. программистов), 1985 г. (оценка) – 5:1.

Исключению посредничества программистов при массовом применении ЭВМ способствовали: увеличение доли недорогих, персонализированных вычислительных средств, не требующих особых условий для своей установки и эксплуатации; абсолютная надежность и замкнутость базовых программных средств при наличии развитого интерфейса с пользователем, позволяющего использовать средства в режиме черного ящика; обязательное наличие технических средств, обеспечивающих диалог с пользователем с использованием текста, символов, изображений и функциональных клавиш; возможность вести диалог с использованием элементов естественного языка и проблемно-ориентированной мнемоники; тщательное проектирование ППП с учетом всего спектра условий их применения; придание пакетам способности адаптируемости путем опроса пользователя и параметрической настройки; представление пользователю возможности формулировать задачу в терминах предметной области с последующим автоматическим выбором метода решения и синтезом программ; адаптируемость ППП к росту квалификации пользователя по мере накопления его опыта; внедрение средствами общего образования основных знаний в области программирования и применения ЭВМ в широком кругу пользователей (всеобщая "компьютерная" грамотность).

Рассмотрим теперь конкретизацию отмеченных общих проблем в сферах науки, образования, технологии и производства.

НАУЧНЫЕ ПРОБЛЕМЫ

1. Создание методов программирования для новых архитектур ЭВМ и новых массовых форм применения ЭВМ

1.1. *СуперЭВМ.* Новые методы необходимы в связи с высокой степенью параллелизма, использованием новых моделей вычислений, применением спецпроцессоров и специальных приемов ускорения вычислений.

1.2. *Спецпроцессоры.* Новые методы необходимы в связи с использованием нетрадиционных моделей вычислений и специальных приемов ускорения вычислений.

1.3. *Программируемые СБИС.* Новые методы необходимы в связи с использованием нетрадиционных моделей вычислений, своеобразным способом представления программы, необходимостью решать задачу геометрического размещения связей по информации и управлению.

1.4. *Персональные ЭВМ.* Новые методы необходимы в связи

со своеобразной организацией программного обеспечения, замыкающегося на индивидуального пользователя, диалогового характера использования ЭВМ, высокой степени адаптируемости к нуждам пользователя.

1.5. *Встроенные микропроцессоры.* Новый вид массового применения ЭВМ, имеющий свои корни в АСУТП и характеризующийся необходимостью сочетать в непрерывной работе ЭВМ вычислительную обработку информации с управлением исполнительными механизмами машины в режиме реального времени. Дополнительная особенность – крайнее разнообразие применений и широкое использование кросс-систем.

1.6. *Распределенная обработка.* Новые методы вызваны потребностью решения проблемы синхронизации, динамического распределения ресурсов, сохранения информации и поддержания живучести.

1.7. *Программирование роботов.* Новые методы вызваны массовостью применения, синтетическим характером задач, необходимостью решения задачи видения, движения и транспортировки, выполнения рабочих функций и поддержания гомеостаза, необходимостью настройки и обучения.

2. Создание методов систематической разработки надежных и высококачественных программ

Рассмотрим компоненты этой проблемы.

2.1. *Доказательное программирование,* при котором построение программы сопровождается доказательным рассуждением, подтверждающим соответствие полученной программы заданным спецификациям задачи.

2.2. *Формальные преобразования программ* как способ улучшения качества программ без нарушения их правильности.

2.3. *Языки спецификаций* как новые формальные системы для записи условия задачи, допускающие систематический вывод программы, удовлетворяющей данным условиям.

2.4. *Теория проблемных областей* как совокупность строго сформулированных общих утверждений об области применения ЭВМ, используемых для синтеза программ и доказательства их правильности.

2.5. *Базы знаний* как организационная совокупность фактов и утверждений о предметной области и о решаемой задаче, представимая в машинной форме и служащая источником приемов, утверждений и данных, используемых при формулировании спецификаций, при синтезе

и оптимизации программ.

2.6. *Методы приближенной оценки правильности программ (отладка и испытания),* позволяющие с заданной степенью риска оценивать правильность программы на основе наблюдений за ее поведением.

2.7. *Методы оценки и измерения качества и надежности программных изделий.*

3. Создание методов автоматического программирования в классах задач и теорий

Эта проблема включает следующие компоненты.

3.1. *Поиск и описание классов задач,* допускающих универсальный метод решения, и параметризация задач из класса, позволяющая решать их путем конкретизации универсального метода или автоматического выбора наиболее подходящего метода из нескольких возможных.

3.2. *Поиск и описание классов задач,* допускающих автоматический синтез программы, отправляясь от формальной спецификации и базы знаний предметной области.

4. Создание научных основ взаимодействия человека и ЭВМ

Для решения этой комплексной проблемы работу необходимо строить по следующим направлениям.

4.1. *Психология человека и ЭВМ.* Должны изучаться проблемы, возникающие при рассмотрении взаимодействия человека и ЭВМ как партнерства человека и автоматического устройства, обладающего антропоморфным поведением. Результаты этих исследований применяются к проектированию схем взаимодействия человека и ЭВМ, в наибольшей степени способствующих созданию комфортных условий для человека и повышению эффективности применения машины.

4.2. *Теория деятельности и решения задач.* Это существующее направление в психологии должно быть распространено на применение ЭВМ. Результаты этих исследований применяются к проектированию схем взаимодействия человека и ЭВМ, прежде всего в интересах создания обучающих систем и систем разработки программ.

4.3. *Организация и средства диалога.* Это комплексное направление должно выработать гамму научно обоснованных технических и программных средств, обеспечивающих быструю

и гибкую организацию диалога при проектировании автоматизированных рабочих мест

и других применений персональных ЭВМ и интеллектуальных терминалов.

4.4. *Непрограммируемое решение задач на ЭВМ.* Это направление должно выработать гамму научно обоснованных технических решений, позволяющих специалисту в предметной области применения ЭВМ, но неспециалисту в области программирования эффективно использовать ЭВМ, оставаясь в рамках привычных ему знаний и действий, применяемых пользователем непосредственно, в режиме непрерываемой предметной деятельности без этапа предварительной формализации условия задачи и указания программы ее решения.

5. Создание базы знаний современного программирования

Речь идет о сборе, фиксации, широкой публикации и внедрении в сферы образования и производства программных средств всей суммы знаний, методов, активное владение которыми позволит создавать высоконадежный и высококачественный программный продукт. Главными компонентами этой базы знаний являются следующие.

5.1. *Руководство программиста.* Оно содержит сумму технических знаний, необходимых для осуществления процесса программирования как профессиональной деятельности: нормативы, стандарты, организационные схемы, перечни и описания технологий, данные по вычислительным средствам и отдельным устройствам, сводку данных из смежных наук – математики, лингвистики, психологии, кибернетики, техники.

5.2. *Теория основных предметных областей.* Дисциплина доказательного программирования базируется на явном и систематическом использовании строго доказанных утверждений, образующих теорию предметной области. Необходимо фиксировать такие теории и представлять их в форме, подходящей для использования в процессе программирования. К ним уже сейчас могут относиться теории таких наиболее массовых областей применения ЭВМ, как математическая теория алгоритмов и программ, вычислительная математика, логика, дискретная математика, теория символьных преобразований – текстов и выражений, вычислительная геометрия и теория обработки изображений.

5.3. *Свод фундаментальных алгоритмов.* По каждому из основных предметных областей должна накапливаться библиотека фундаментальных алгоритмов, присущих данной области. Такие библиотеки сейчас существуют, но их общезначимость сильно принижена реальной невозможностью широкого переноса между разными ЭВМ и погружения в другие обстановки с доказательным сохранением их правильности. Своды фундаментальных алгоритмов должны не только формироваться в машинно-независимой форме, но и содержать проверяемые сертификаты их правильности, т.е. либо доказательство, либо описание полной и воспроизводимой системы испытаний данного алгоритма.

5.4. *Языковые средства программирования.* Должна поддерживаться стабильная система стандартных, т.е. одинаково понимаемых языковых средств программирования на всех уровнях взаимодействия с ЭВМ: программируемые СБИСы, микропрограммы, машинные языки, автокоды и ассемблеры, алгоритмические языки, языки операционных систем, языки управления и манипулирования данными, языки диалогового взаимодействия с ЭВМ, языки спецификаций, входные языки прикладных пакетов.

Каждый языковой стандарт должен придавать единое и общезначимое толкование не только собственно объектам языка, но и элементам внешней операционной обстановки, взаимодействующей с языковыми объектами. Такими внешними интерфейсами, в частности, являются: способы выполнения арифметических операций, системы команд, протоколы сопряжений с магистральными шинами, сетями и линиями связи, протоколы аварийных прерываний, примитивы операционной системы, примитивы системы управления базами данных.

Стандартизация внешних интерфейсов наряду со стандартизацией языковых средств позволит унифицировать программные средства, т.е. применять их многократно без изменения в разных операционных обстановках, в том числе на разных ЭВМ с сохранением правильности их функционирования.

Со временем должна сложиться некоторая общая и общепринятая языковая среда, поддерживающая все языковые уровни программирования, средства манипулирования с программами и способы точного рассуждения о них, а также средства перехода с одного уровня на другой (лексикон программирования). Лексикон должен стать основным средством выражения базы знаний программирования.

ПРОБЛЕМЫ ОБРАЗОВАНИЯ

1. Формирование вузовской специальности "программирование"

Со временем должна быть организована подготовка по вузовской специальности "программирование" со специализациями: теоретическое программирование (университет), системное программирование (университет, втуз), прикладное программирование (втуз).

На подготовку по этой специальности должна быть направлена часть контингента, принимаемого ныне на специальность "прикладная математика" со специализацией "математическое обеспечение ЭВМ" и "математическое обеспечение АСУ", а также некоторые родственные втузовские специальности и, кроме того, часть нового контингента, возникающего либо благодаря расширению приема в вузы, либо из-за сокращения подготовки по другим, менее дефицитным, специальностям.

Необходимость создания этой специальности, особенно с учетом перспективы, вызвана следующими причинами.

– Складывающаяся невозможность реально повысить уровень образовательной и профессиональной подготовки программистов в рамках единого плана по специальности "прикладная математика". Грубо говоря, в университетах не доучивают программированию, а во втузах – прикладной математике.

– Несколько иная методологическая установка специальности "прикладная математика", ориентирующая скорее на решение задач, т.е. постановку задачи и использование ее результатов. При таком подходе специалист по прикладной математике становится скорее специалистом по применению ЭВМ, привыкая смотреть на программирование как на промежуточный этап в решении задачи и не вырабатывая в себе вкус к процессу программирования как к таковому, ведущему к созданию программного продукта. Такая установка, хорошо известная у выпускников университетов, объективно препятствует становлению программирования как профессии.

– Пользователь, решивший задачу, забывает о программе. В то же время для профессионального программиста завершение составления программы является лишь отправной точкой принятия на себя полной ответственности за поведение ЭВМ, вооруженной его изделием. Эта повышенная социальная ответственность, приближающая программистов к профессиям с развитым этическим кодексом (например, врачам), также требует особого подхода к их обучению.

2. Постановка вузовского курса современного программирования

В рамках формирования базы знаний современного программирования должна быть тщательно отобрана сумма знаний и умений, предвидимых в качестве фундаментальных знаний на перспективу, и положена в основу:

1) нового курса общего программирования объемом порядка 200 часов плюс практикум, предназначенного для всех специальностей, связанных с применением ЭВМ;

2) полного курса основ программирования объемом порядка 600 часов плюс практикум, плюс два проекта, предназначенного для подготовки профессиональных программистов.

Эти курсы в экспериментальном порядке должны быть прочитаны в ближайшие годы в десятке вузов страны, а после конкурса программ и учебников внедрены в регулярную систему преподавания в конце XII пятилетки.

3. Внедрение ЭВМ и преподавания информатики в среднюю школу

Со временем должно получить широкое распространение применение ЭВМ в школьном учебном процессе на базе кабинетов информатики, оборудованных персональными ЭВМ, объединенными в локальную сеть. Темпы этого внедрения будут определяться многими факторами технического, экономического и организационного характера, однако в любом случае нужно добиваться их распространения, настолько широкого, чтобы оно могло решить задачу создания среднеобразовательной специальности программиста (см. ниже), а также создать необходимый подпор знаний в области информатики среди контингента, поступающего в вузы по специальностям, тесно связанным с применением ЭВМ.

В школах, снабженных кабинетами информатики, должны сложиться потоки, в которых использование ЭВМ начинается с младших классов, вводный курс информатики преподается в пятых–восьмых классах, а в старших – доступно регулярное применение ЭВМ в учебной работе и труде, а также углубление знаний информатики в предпрофессиональной подготовке.

В двенадцатой пятилетке должны быть проведены подготовительные эксперименты, которые позволят организовать выпуск десятых классов с такими компьютеризованными потоками в заметных количествах.

4. Формирование среднеобразовательной специальности по программированию

Широкое развитие профессионального программирования, гибких автоматизированных производств и автоматизированных рабочих мест требует создания среднеобразовательной специальности техника-программиста и рабочих профессий оператора-наладчика ГАП и АРМ. Подготовка должна частично опираться на проникновение ЭВМ в общую среднюю школу, а частично обеспечиваться собственными учебными программами по линии техникумов и профессионального образования.

5. Профессиональная переподготовка

По мере формирования базы знания современного программирования и методов применения ЭВМ (особенно микропроцессоров, встраиваемых в машины) необходима массовая переподготовка всего контингента системных и прикладных программистов и инженеров-конструкторов.

Поскольку в связи с большим объемом переподготовки и с динамическим характером самого вычислительного дела процесс переподготовки будет носить постоянный характер, уже в ближайшие годы необходимо создать постоянную сеть учебных центров по вычислительной технике и программированию общей пропускной способностью примерно в 100 тысяч учебных мест.

ПРОБЛЕМЫ ТЕХНОЛОГИИ

Создание технологии промышленного производства программных средств и ее быстрое внедрение в практику программирования являются важнейшей задачей на пути полноценного применения вычислительной техники. Для этого необходимо решить следующие подзадачи.

1. Создание организационной структуры разработки программных средств как продукта производственно-технического назначения

Разработка программных средств обладает многими особенностями, однако для более быстрого продвижения вперед целесообразно в качестве первого приближения к ее организационной структуре взять зрелую и отшлифованную многими десятилетиями организационную структуру проектирования и

изготовления изделий машиностроения, поскольку и большие программы, и программно-аппаратные комплексы, и машины подчиняются одним и тем же законам общей системотехники.

2. Отбор методологий разработки новых программ

К настоящему времени теория и практика программирования предлагают достаточно богатый набор методов систематического построения программ. К ним, в частности, относятся: пошаговое уточнение, структурно-логический синтез, трансформационный синтез, R-технология, HIPO-технология.

Задача состоит в том, чтобы построить развернутый теоретический базис каждой методологии, придать ей систематический характер, найти для нее адекватное внешнее представление и погрузить в выбранную организационную дисциплину разработки программ.

3. Повторное использование программных разработок

Необходимо разработать и внедрить в массовую практику методы, обеспечивающие многократное использование "деталей" и "заготовок" программных изделий-модулей, процедур и универсальных программ.

Для модулей и процедур нужно будет решить проблему сборочного программирования – систематического или автоматизированного процесса сборки программного средства на основе методов модульного программирования, теории абстрактных типов данных и методов автоматического программирования.

Для универсальных программ и многоцелевых пакетов следует решить проблему конкретизирующего программирования на основе методов смешанных вычислений, микрообработки и текстового редактирования.

4. Создание технических средств разработки программ

Речь идет о создании в XII пятилетке профессиональной ПЭВМ нового поколения, представляемой в качестве автоматизированного рабочего места программиста. Самые общие ее характеристики:

- скорость порядка 1 млн опер./с;
- емкость ОЗУ порядка 1 мегабайта;
- сменные гибкие магнитные диски для загрузки;
- компактный жесткий диск типа "Винчестер" емкостью порядка 10 мегабайт;
- стандартный интерфейс для объединения в локальную сеть и для связи с центральной ЭВМ;
- графический дисплей на 1 млн элементов изображения;
- вывод с программируемыми шрифтами, качества не ниже пишущей машинки и с возможностями двумерной графики.

5. Создание интегрированных систем разработки программ (ИСРП)

Уже упоминавшиеся ИСРП должны создаваться в две очереди:

- 1) все выпускаемые типы ЭВМ должны иметь развернутые на доступных вычислительных средствах ИСРП первого поколения, поддерживающие дисциплину и технологию программирования в ассортименте выбранных алгоритмических языков, не претендуя на методологию доказательного программирования или на автоматизацию сборочного и конкретизирующего программирования;
- 2) одновременно должны отрабатываться прототипы ИСРП второго поколения, развернутые на перспективных вычислительных средствах и охватывающие указанные выше методы доказательного, сборочного и конкретизирующего программирования.

6. Создание технологических стандартов

На все основные стадии проектирования, изготовления, сопровождения и изъятия программного продукта должны быть разработаны, апробированы и внедрены в ближайшие годы технологические стандарты.

ПРОБЛЕМЫ ПРОИЗВОДСТВА И РАСПРОСТРАНЕНИЯ

В этом разделе концентрируются главным образом кадровые и организационные проблемы промышленного производства программных средств, которые со временем должны найти свои решения в нормативах и законодательных актах.

1. Установление профессионального статуса программирования

– Установить должностную структуру разработчиков и руководителей разработки программных средств, а также систему должностных окладов, средств поощрения и надбавок, отвечающих категории сложности программных средств и квалификации (классности) программистов.

– Разработать систему периодической государственной сертификации (аттестации) программистов на предмет их соответствия характеру и степени ответственности выполняемой работы.

В основу этой системы положить постоянно развивающийся перечень квалификационных требований, который во время аттестации сопоставляется с результатами экзаменов и послужным списком, характеризующим объем и качество разработанного программного продукта.

– Определить порядок перевода лиц, занимающихся разработкой программ, но находящихся на других должностях, на новые должности, отвечающие статусу профессионального программирования. В системе новых должностей должны быть должности внеклассных программистов в качестве буфера подготовки классных программистов и для привлечения программистов, не связанных с разработкой промышленного программного продукта.

– Определить этический кодекс профессиональных программистов, отражающий повышенный уровень социальной ответственности программиста за поведение машин и устройств, управляемых разработанными им программами.

2. Организация центров и подразделений по производству программного продукта

– Следует произвести начальный учет организаций и подразделений, выпускающих программные средства, и осуществить перепись программистов, позволяющую получить исходные данные о наличном корпусе специалистов.

– Определить головные специализированные организации по производству программных средств.

– Найти формы взаимодействия с рассеянными подразделениями профессиональных программистов (в вузах, академических институтах, проектных организациях), обладающих хорошим заделом, но неспособных самостоятельно производить программные средства по полной технологической цепочке. Эти организационные формы должны способствовать концентрации труда квалифицированных системных и прикладных программистов.

– Найти организационные и технологические формы производства экспериментальных программных средств и прототипов программных изделий, способствующие скорейшему воплощению новых идей и изобретений в серийный программный продукт.

– Разработать систему учета производства и распространения программного продукта, позволяющую располагать полной информацией об объеме производства, производственном потенциале предприятий, производительности программистов, показателях надежности и качестве, степени массовости применения программного продукта.

3. Организация сети по распределению, продаже и сопровождению программных средств

- На базе фондов алгоритмов и программ должны быть образованы хозрасчетные предприятия по тиражированию, распространению и продаже программных средств, в особенности массового производства.
- Разработать систему мер, способствующую повышению тиражности программных средств, обладающих высоким качеством. Эта система мер не должна тормозить появление аналогичного программного продукта, обладающего повышенными потребительскими свойствами.
- Разработать разветвленную и действенную систему гарантийного обслуживания программных средств, своевременного учета рекламаций или сообщений об обнаруженных ошибках, изъятия ошибочных или устаревших изделий и их замены новыми версиями. Любое программное средство, проданное или поставленное как товар, должно в любой момент времени иметь известную организацию, юридически ответственную за поведение программного продукта и реализующую соответствующие гарантии.
- Создать сеть розничной продажи программных средств массового применения, в особенности для персональных ЭВМ, применяемых, прежде всего, в быту, в процессе образования и интеллектуальной деятельности.