

строения системного обеспечения пакетов прикладных программ // Пакеты прикладных программ. Системное наполнение. М.: Наука, 1984. С. 12-28.

3. Васильев Ф.П. Численные методы решения экстремальных задач. М.: Наука, 1980. 518 с.

4. Карманов В.Г. Математическое программирование. М.: Наука, 1980. 256 с.

5. Тетерев А.Г. Методы одномерной оптимизации: Метод. указания. Куйбышев. 1983. 45 с.

6. Трамбле Ж., Соренсон П. Введение в структуры данных. М.: Машиностроение, 1982. 784 с.

УДК 681.142.2

А.Н.Ковшов

СРЕДСТВА УНИФИКАЦИИ ПРОГРАММНО-МЕТОДИЧЕСКОГО
ОБЕСПЕЧЕНИЯ АУИК

(г. Свердловск)

Внедрение вычислительной техники в процесс подготовки специалистов в области информатики и вычислительной техники осуществляется за счет создания проблемно-ориентированных автоматизированных учебно-исследовательских комплексов и интеграции их в локальную вычислительную сеть.

Обеспечение процесса обучения и исследования на базе готовых пакетов и программно-аппаратных компонент, а также развития автоматизированного учебно-исследовательского комплекса (АУИК) осуществляется системой программно-методических модулей (ПММ). ПММ содержат три равнозначных составляющих: программное обеспечение, программную документацию и методическое обеспечение.

Широко распространено заблуждение в том, что программное обеспечение (ПО) ограничивается рамками исходных текстов программы. Документация на ПО разрабатывается во вторую очередь и ей не уделяется столь пристальное внимание как созданию ПО. Здесь есть определенный парадокс. Реально, не имея программной документации (или

обладая неадекватной и неполной документацией), невозможно воспользоваться программным обеспечением. С другой стороны, если есть отдельно сделанная документация на программную систему, может не составить особого труда изготовить эту систему. Тем более становится ясно, что это так, если учесть, что основная трудоемкость создания сложного ПО состоит не в собственно программировании, а в проектировании, т.е. разработке функций системы и определении принципиальных аспектов реализации. В хорошей программной документации именно эти моменты и находят отражение. Отсюда следует и то, что первоначальная версия документации должна создаваться до разработки программ.

Отладка ПО сопровождается отладкой документации. Прослеживается определенная аналогия между ПО и документацией. Полнота этой аналогии делается еще заметней, если рассмотреть выполнение программ не на отдельно взятой ЭВМ, а в рамках системы человек-машина. Программа управляет машиной, документация "управляет" человеком.

Однако нельзя в полной мере утверждать последнее. Программная документация скорее дает набор не предписаний, а возможностей. Как ими воспользоваться — дело того, кто эксплуатирует ПО. В достаточно сложных, богатых средствами системах вопрос освоения и правильной эксплуатации выливается в проблему.

Средства есть, но как правильно работать с этими средствами? Всякий пользователь системы, осознавший этот вопрос, попытается найти в системе соответствующие методические рекомендации. Как правило, его ждет разочарование, так как созданию методического обеспечения уделяется еще меньшее внимание, чем разработке программной документации.

Методическое обеспечение (МО) должно объяснить, в каких ситуациях какие средства и как следует использовать для достижения заданных целей. Любые формы МО приемлемы, если они отвечают этим требованиям.

В первом приближении МО можно представить как набор регламентирующих документов, например таких как описание применения; как оформлять программу и методики испытаний ПО; как планировать конфигурацию файловых систем, осуществлять их проверку и восстановление, наладить систему поколений копий данных; как действовать в случае модификации программ общего пользования, в случае обнаружения ошибок и неточностей и т.д.

Ясно, что это очень большая система, наполненная массой конкретных и иногда частных сведений. Отсюда следует, что, во-первых, ее невозможно сделать заранее. Такая система МО может появиться в законченном виде разве что в качестве окончательного продукта. Во-вторых, необходимо предусмотреть, чтобы система МО действительно являлась системой, а не однородным набором сведений, в противном случае ею нельзя будет пользоваться. Поэтому система МО должна быть открытой и структурированной.

Методическое обеспечение, как и другие компоненты ПММ, проходят процесс отладки, модификации и расширения функций. Существует глубокая аналогия между МО и программным продуктом. Все эти компоненты, помимо ориентированности на общую цель, обладают одинаковыми свойствами — тенденцией к отладке, развитию и иерархической структурой. Попытка рассмотреть МО обособленно приведет в лучшем случае к созданию искусственного бюрократического комплекса регламентаций, неспособного учесть реальные потребности и эксплуатации сети АУИК. Традиционный подход к созданию МО как раз характеризуется такими попытками. В этом случае фиаско методического обеспечения вполне закономерно, и уже не вызывает удивления безрезультатность обращения пользователя к системе за методическими рекомендациями.

Поскольку объективно существует единство компонент программного продукта, естественно требование единства форм представления ПО, МО и программной документации на всех уровнях разработки и эксплуатации АУИК. Единство формы автоматически влечет за собой необходимость единых средств и процедур для поддержки компонент программного продукта.

Поскольку ПММ прежде всего выступает как система документов, то возникает намерение реализовать ПММ как данные и применить систему, позволяющую использовать данные в качестве управляемого ресурса. Существует относительно новое направление, так называемые системы словарей-справочников данных (СССД) /1/, которые предлагают интересные решения соответствующих вопросов. К сожалению, в стране СССР не получили практически никакого развития, и СССР остается рассматривать только как отдаленную перспективу. Реально можно ориентироваться только на те средства, которые включены в состав базового варианта СПО АУИК.

Среди операционных систем СПО АУИК особое место занимает *UNIX* (адаптированные варианты - ДЕМОС, МНОС, ИНМОС). Файловая система *UNIX* иерархически структурирована и фактически служит базой данных программного проекта. *UNIX* обладает мощными средствами обработки текстовых данных. Необходимо подчеркнуть также простоту и удобство использования системы, повышенный сервис пользователя. Наконец, *UNIX* - мобильная система, существующая на всех типах современных ЭВМ (от мини- до суперкомпьютеров). В *UNIX* имеется мобильный сетевой пакет. Все это заставляет рассматривать *UNIX* как основную систему поддержки ПММ в интегрированной сети АУИК.

Концепции, положенные в основу *UNIX*, привели к тому, что эта ОС получило ярко выраженное свойство систематичности. Проблемные разработки выполняются теми же средствами, с помощью которых была создана сама ОС. При правильном (в методическом смысле) использовании этих средств проблемной разработке придается стандартная для ОС форма. Тем самым система ПММ естественным образом вливается в состав ОС и переходит в статус системного обеспечения. Что бы ни было сделано, все с помощью *UNIX* может быть оформлено частью самой ОС *UNIX*. Таким образом, форма проблемного программного продукта унифицируется, причем способом, принятым для самой операционной системы.

Если в СПО АУИК рассматривать *UNIX* как средство реализации и поддержки системы ПММ АУИК, то возникает возможность не создавать для поддержки ПММ никакого специального обеспечения. Те же самые средства разработки, которые применяются для создания ПО и документации, могут использоваться для создания и развития МО.

Структурированность и способность к развитию методического обеспечения (равно как и других компонент программного продукта) базируется на структурированности и открытости *UNIX*. Следует особо отметить интерактивную обучающую систему *LEARN OS UNIX* как средство, полезное для поддержки МО. Эта система позволяет придать МО не только форму иерархического комплекса документов, но и "оживить" методики. В целом разработка программного продукта подчиняется принципам систематического программирования (метод нисходящего проектирования).

Если рассмотреть преломление этих принципов в средства *UNIX*, то в общих чертах процесс создания и развития ПО предстает в виде следующей схемы.

Вся работа происходит в рамках сети АУИК. Каждый АУИК имеет собственную иерархическую файловую систему. Особое место в сети занимает комплекс, который назовем ФОНД. Здесь концентрируется в конечном итоге разработанные ПММ. Отсюда они тиражируются среди пользователей. Файловая система комплекса ФОНД содержит подката - логи */PRO/BIN* - для хранения готового ПО, */PRO/SYS* - для исходных текстов ПО, */PRO/DOC* для программной документации и методического обеспечения, */PRO/MAN* - для оперативной документации по разработанным системам. Перечисленные каталоги структурированы по различным системам или подсистемам этих систем.

В каталоге */PRO/DOC* в общем случае содержатся следующие документы (или группы документов) по разработанной программной системе: описание применения - основные функции, свойства, области применения системы;

описание системы - описание изобразительных средств системы, например, описание языка программирования (система-транслятор), описание функций программ (система - пакет программ). В этом документе достаточно полно и точно описаны возможности средств системы для того, чтобы пользователь мог практически эти средства применять;

руководство программиста - по сути это руководство по эксплуатации, поясняющее форматы вызова (запроса) используемых средств. Например, описание команды вызова транслятора и список возможных диагностик;

руководство системного программиста - описание более тонких свойств системы, общих моментов реализации, принципиальных технических решений, положенных в основу реализации и главных внутренних структур данных. Например, описание форматов таблиц устройства в ядре операционной системы, описание механизма планирования процессов;

руководство по генерации - описание постановки системы в конкретной операционной среде, способов генерации системы из исходных текстов для придания системе нужных свойств и характеристик;

контрольный пример - описание функций и способа запуска контрольного примера, проверяющего правильность работы основных средств системы;

программа и методика контроля и испытания – описание состава проводимых проверок и испытаний и порядок их проведения при приемке системы;

методические рекомендации – описание порядка и приемов использования системы в различных ситуациях, рекомендации по комбинированию средств системы, примеры стереотипов, примеры использования системы для решения типичных задач.

Поскольку СПО АУИК само является программной системой, то в каталоге */PRO/DOC* содержится соответствующая документация, включая методические руководящие материалы по применению СПО. Каталог */PRO/MAN* содержит оперативную документацию по программным системам, в концентрированной форме дающей функции основных средств. Эта документация не претендует на полноту описания, а скорее ориентирована на подсказку – напоминание в процессе работы пользователю, уже овладевшему системой.

Для форматизации и компоновки этих каталогов *UNIX* имеет соответствующие средства. Функциями фонда являются: приемка разработанного ПО (включая программы, документацию и методические материалы); документальное оформление приемки; информирование по службе новостей сети заинтересованных пользователей об изменениях фонда; оформление запроса на передачу ПО пользователям; передача ПО.

В отличие от практикуемых фондов алгоритмов качественно новым свойством фонда, работающего в составе сети АУИК, является оперативность доступа и модификации, так как практически все функции доступа в фонд могут быть выполнены в рамках сети.

Интегральная система АУИК по моделированию организационно-проектных ситуаций может быть использована для отработки различных организационно-проектных ситуаций, возникающих на производстве. Рассмотрим один из примеров.

Разработка начинается, когда заказчик (преподаватель) формулирует потребность-заказ на систему. Логично этот заказ зафиксировать в каталоге фонда (допустим, */PRO/REQ*). Используя сеть связи, нетрудно это сделать оперативно. Разумеется, заказ с помощью текстовых редакторов и форматтеров приводится к стандартной для СПО форме. По службе новостей сети фонд информирует исполнителей (обучаемых) о заказе. Исполнитель может, воспользовавшись услугами

службы информации ФОНДа, узнать много полезного. Например то, что аналогичная система уже разработана или разрабатывается.

Вступив в отношения с преподавателем, обучаемый в общем случае совместно с преподавателем прежде всего создает функциональные спецификации (ФС) системы, т.е. документ "описание системы" и согласует его. Взаимодействие преподавателя и обучаемого поддерживается почтовой службой сети.

Размещать "описание системы" в ФОНДе еще рано — это только первоначальный вариант, который будет уточняться в процессе разработки. Фактически, это не что иное, как техническое задание на разработку ПО. В своей файловой системе обучаемый организует по аналогии с фондом личные каталоги *BIN, SYS, DOC* и *MAN*, размещая описание системы в *DOC*.

На следующем этапе обучаемый приступает к созданию системы. Здесь самое время создать в *DOC* "Руководство системного программиста", зафиксировав принципы реализации системы. Затем создается и отлаживается верхний уровень системы, причем исходные тексты фиксируются в каталоге *SYS*, а готовые к выполнению программы — в *BIN*. Какой-то набор функций системы (быть может очень ограниченный) в первом приближении сделан и отлажен. Также в первом приближении заполнены каталоги *BIN, SYS, DOC* и *MAN* личного фонда. Дальнейшая реализация должна быть продолжена развитием подсистем, пока еще ограниченных в своих возможностях или еще вообще не сделанных и не подключенных к системе.

На этой стадии обучаемый переходит в роль заказчика. Он устанавливает контакт с субисполнителем, формулирует свой заказ (в каталоге *REQ* личного фонда), передает субисполнителю копию (фрагмент) своего личного фонда, создает с субисполнителем требуемые ФС, и вся схема разработки повторяется на уровне субисполнителя. Смена роли в обучении необязательно сопровождается сменой лиц, исполнитель может выступать в роли заказчика для самого себя. Исполнитель обладает большой свободой при создании ПО. Он может компоновать для размещения разрабатываемого продукта файловые системы любой конфигурации, использовать все доступные средства и формы представления данных, но в момент окончания работы обязан привести результат к стандартному виду, который регламентируется методическим обеспечением СПО АУИК.

Заканчивается работа приемкой результатов от обучаемого в ФОНД.

После тиражирования и опытной эксплуатации ПО ФОНД накапливает сведения о замеченных ошибках и предложения по совершенствованию ПО, допустим в каталоге *PRO/BUGS*, который в этот период отслеживается исполнителем.

Приведенную методику работы следует рассматривать как общую скелетную схему. Реальные методические материалы СПО АУИК должны быть более конкретизированы, но общая канва системы МО как руководство к действию останется неизменной. Изменение конкретных форм МО в данном случае не только допустимо, но и желательно, так как это единственный способ совершенствования ПММ до нижнего уровня, которое может быть выполнено только совместными усилиями функционеров в сети. Для этих целей ФОНД собирает предложения по совершенствованию ПММ в каталоге *PRO/IMP*.

Из всех операционных систем, входящих в состав СПО АУИК, *UNIX* наиболее полно отвечает смыслу термина "система", являясь именно системой, а не суммой средств. *UNIX* в большей степени, чем другая ОС, ориентирована на коллегиальную деятельность и электронную обработку текстов. Эти обстоятельства заставляют ФОНД погрузить в среду *UNIX*. Возникает вопрос, в какой мере СПО ФОНД пригодно и доступно в других ОС СПО АУИК.

В техническом аспекте ответ положителен. Принятая структура интегральной системы АУИК такова, что *UNIX* выполняется на центральной ЭВМ комплекса. Другие ОС (РАФОС, ОС РВ и любые другие) расположены в терминальных ЭВМ, радиально подключенных к центральному узлу. Суть центрального узла состоит в том, что именно эти узлы объединяются в *UNIX*-сеть, которая поддерживается сетевыми средствами ОС *UNIX*. Терминальные узлы имеют выход в сеть и вообще полный доступ к средствам *UNIX* центрального узла посредством прозрачного режима, который может быть установлен из любой ОС терминальной ЭВМ. Более того, терминальным узлом может быть не отдельная ЭВМ, а целая сеть иной природы. Таким образом, доступ к системе ПММ можно получить с любого терминала сети АУИК.

Положительное решение существует в методическом аспекте. Как уже упоминалось, исполнитель имеет практически неограниченную свободу выбора средств, и здесь необходимо подчеркнуть, что эта сво-

бода распространяется вплоть до выбора любого языка и любой ОС. Единственное требование, которое должен выполнить исполнитель, — привести результаты работы к стандартной для ФОНДа форме. Это всегда можно сделать, так как существует прозрачный режим работы терминальной ОС. Более того, подготовку необходимых средств можно выполнить с помощью средств обработки текстов терминальной ОС. Во-первых, для некоторых ОС (РАГОС) существуют совместные с *UNIX* форматтеры. Во-вторых, для целей хранения и распространения программного продукта необязательно придавать ему форму *UNIX* системы, если этот продукт предназначен для использования в другой операционной среде. В функции ФОНДа входит централизация, координация и методическое обеспечение работ, а вовсе не использование ПО.

Библиографический список

1. Леон-Хонг Б., Плагман Б. Системы словарей-справочников данных. М.: Финансы и статистика. 1986. 311 с.

УДК 681.3.06

С.И.Борисов, Б.С.Мищенко, Л.В.Попов

СИСТЕМА АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ БИОФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА

(г. Ленинград)

Специфика биофизических экспериментов, проводимых в вузе, заключается в наличии большого числа относительно несложных установок, позволяющих реализовать разнообразные способы исследований биологических объектов. В состав технических средств входят микроЭВМ, аппаратура КАМАК, а также приборы, имеющие собственные интерфейсы сопряжения с ЭВМ. Возможности автоматизированного эксперимента в значительной степени определяются свойствами прикладного программного обеспечения (ПО), объем которого неизбежно растет. Поэтому остро стоит проблема снижения трудоемкости создания систем автоматизации биофизических экспериментов, повышения надежности и