

Б. С. ПАДУН

ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ПОДГОТОВКИ ПРОИЗВОДСТВА

Рассматриваются основные положения технологии построения программных комплексов, проектирования алгоритмов и программных компонентов, организации информационного взаимодействия программных комплексов и компонентов, формирования схем решения технологических задач.

Ключевые слова: технология проектирования алгоритмов и программ, программное обеспечение, алгоритмический модуль, программный модуль, инструментальная система.

Введение. Система автоматизированного проектирования (САПР) технологической подготовки производства (ТПП) включает большое число составных разнородных элементов и имеет сложное управление, ее компоненты имеют разный уровень формализации, ее пользователи — это специалисты в различных областях знания, при решении задач используются разные (порой противоречивые) критерии, система может допускать ошибки, ей необходимо адаптироваться к внешним условиям и развиваться во времени.

Когда решаются локальные задачи технологического проектирования или когда они решаются только в диалоговом режиме, разработчик практически не задумывается о технологии проектирования алгоритмов и программных компонентов. Но переход к созданию интегрированной системы сразу „вскрывает“ проблемы проектирования, вызванные отсутствием такой технологии.

Цель настоящей статьи — сформулировать основные положения технологии проектирования САПР ТПП [1, 2].

1. *Формирование состава и структуры программного обеспечения (ПО) САПР ТПП.* В работах [2, 3] САПР ТПП определена как совокупность программных, информационных, лингвистических и технических слоев. Программные слои обеспечивают связь пользователя с абсолютными ресурсами. Такой подход известен как подход Е. Дейкстра. В работах [2, 3] предложена организация связи САПР ТПП с системами управления базами данных через специальный слой, который был назван интерфейсным. Связь пользователя и программных компонентов обеспечивается специальными языковыми средствами (лингвистическими слоями).

Применение слоевого подхода позволяет решить задачу независимости программного приложения от способа хранения данных и используемых СУБД, создает предпосылки для эффективного использования инвариантных программных средств и повышает адаптивные свойства системы.

2. *Определение структуры и состава пакета и управления пакетом программ.* В работе [4] предлагалось использовать для САПР ТПП два варианта организации пакета: простой для локальных систем и сложный — для комплексных и интегрированных. Состав задач ТПП и последовательность их решения определяются динамически во время выполнения работ. Для этого управляющая программа пакета реализует программную грамматику, где словарь (терминальные символы) — это имена компонентов ПО; классы (нетерминальные символы) — это имена функций ТПП; аксиома — это имя составной части управляющей программы „монитор“, классы „система“ и „аварийное завершение“; правила подстановки — это возможные структурные варианты терминальных и нетерминальных символов, которые разбиты по классам.

В процессе проектирования ТПП некоторые компоненты ПО или пользователь в режиме диалога задают правила подстановки. „Монитор“ реализует предложение, состоящее из

терминальных и нетерминальных символов. Передача управляющих данных выполняется через коммуникационную область памяти, проблемных — через общую область.

3. *Выбор способа построения программного компонента*, решающего конкретную задачу ТПП. В работе [2] предложены правила определения способа построения программного компонента, которые основаны на согласовании классификаций задач проектирования и типов программных компонентов. Задачи разделяются на четыре группы (рис. 1):

— инвариантные по отношению к производственным условиям, где эти задачи решаются, поэтому программные компоненты строятся как неизменяемые (оригинальные);

— с неизменяемым составом параметров, по которым принимаются решения, и схемой решения, но с изменяемыми пороговыми значениями параметров. В этом случае строятся оригинальные программные компоненты, для которых пороговые значения параметров выбираются из базы данных или передаются через формальные параметры;

— с изменяемой схемой решения, но с возможностью многократного использования. Алгоритмы решения этих задач следует реализовывать инвариантными программными средствами, имеющими в своем составе унифицированный набор обрабатывающих программ;

— с изменяемой схемой решения, но с возможностью одноразового использования. Алгоритмы решения этих задач реализуются программными компонентами, которые вызываются пользователем в процессе решения задачи. Для вызова программных компонентов могут использоваться языки, приближенные к естественному языку пользователя, меню или справочники.



Рис. 1

Использование данного положения позволяет создать минимально необходимый и достаточный состав программных и алгоритмических средств, на основе которых строится САПР ТПП, это увеличивает адаптивные свойства последней.

4. *Формирование схемы решения задачи* на основе принципа неокончателных решений. В этом случае схема решения задачи представляется в виде последовательности алгоритмов, где каждый алгоритм решает только часть задачи и определяет допустимое множество решений. Укрупненная модель решения задачи в общем случае представляется в виде

$$\left. \begin{aligned} L \quad K \quad \left(\bigcap_i K_i \quad \Gamma_i \quad O_j \right) &= r_j, \\ \Gamma_i &= \Gamma_{\lambda i} \quad \dots \quad \Gamma_{2i} \quad \Gamma_{1i}, \\ \bigcap_i K_i \quad \Gamma_i \quad O_j &= R_j, \quad r_j \in R_j, \end{aligned} \right\}$$

где L — диалоговый оператор уточнения или корректировки решений; K — глобальный критерий оценки решений; K_i — частный критерий оценки решений i -й задачи; Γ_i — алгоритм решения i -й задачи; O_j — j -й объект, относительно которого принимается решение; r_j — j -е решение; R_j — множество допустимых решений. Наличие членов $\bigcap_i K_i \Gamma_i O_j$ и

$\Gamma_i = \Gamma_{\lambda_i} \dots \Gamma_{2i} \Gamma_{1i}$ говорит о том, что процесс принятия решений может быть соответственно параллельным и последовательным. Алгоритмы Γ_{λ_i} могут быть реализованы путем синтеза, выбора и поиска решений, порождения новых множеств, вычисления и т.д., которые являются инвариантными по отношению к конкретным алгоритмам.

5. Унификация представления алгоритмического и программного модулей САПР ТПП. Алгоритмический модуль (АМ) в общем случае включает блоки, представленные на рис. 2, а. Назначение блоков понятно из их названия, исключение составляет блок б. Он описывает действия, направленные на изменение условий проектирования, к которым автор относит те свойства информационных объектов, которые изменяются после применения выбранного решения. Например, во время проектирования технологических процессов после применения выбранного инструментального перехода изменяется состояние заготовки. При возможности изменения условий проектирования после выбора решения работа алгоритмов становится независимой от предыстории развития процесса проектирования. Работа алгоритма — это самостоятельная и независимая часть процесса проектирования.

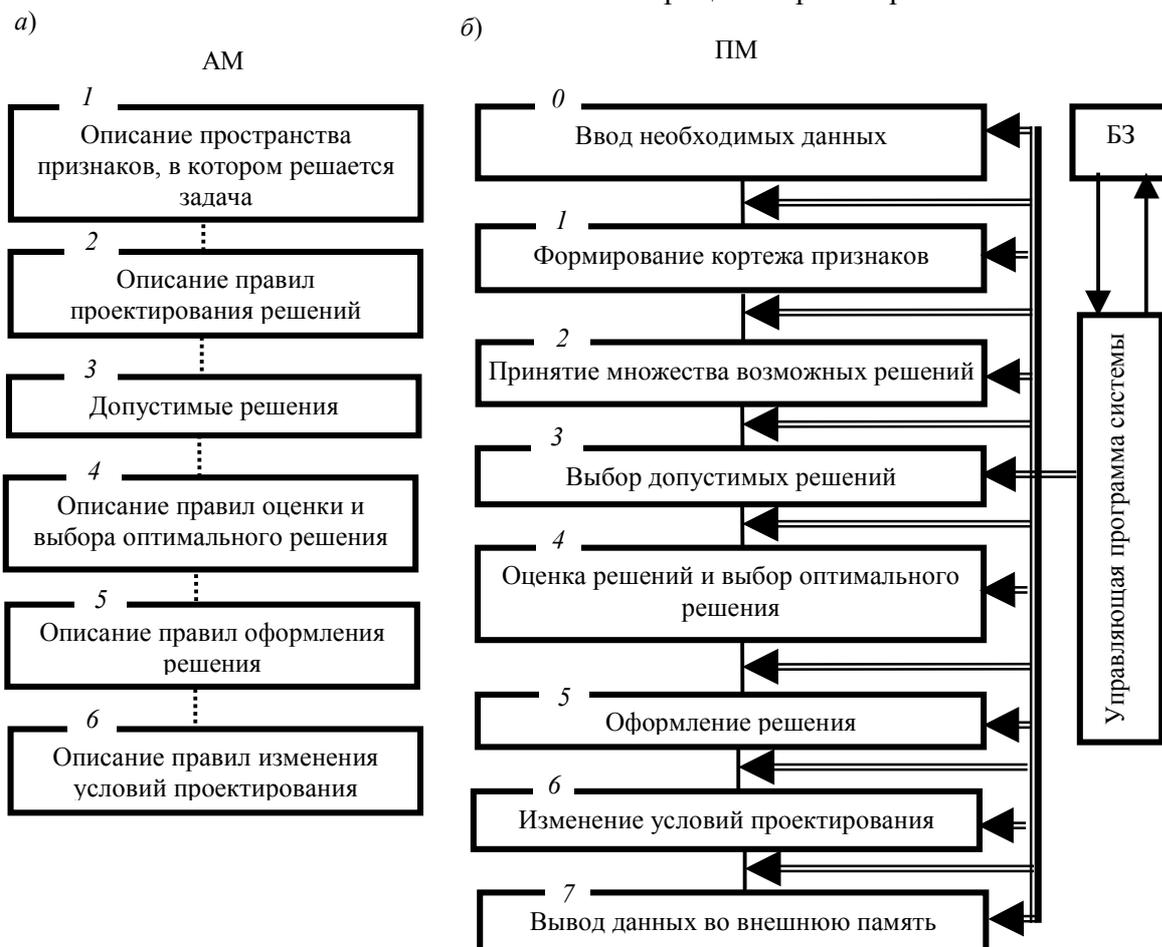


Рис. 2

Заметим, что если в блоке 2 будут описаны правила, позволяющие выбрать допустимые (т. е. не зависящие от производственных условий) варианты решений, то алгоритм, кроме блока 3, может тиражироваться. Адаптация к конкретным производственным условиям произво-

дится заменой блока 3. Следовательно, можно создать библиотеку алгоритмов, которые следует рассматривать как хранилище универсальных технологических знаний. При этом упрощается процесс развития алгоритмов, их обслуживания, распространения и описания.

Программный модуль (ПМ) имеет сходную с АМ структуру (см. рис. 2, б), за исключением первого (0) и последнего (7) блоков, которые необходимы для ввода—вывода данных. При формировании пространства признаков могут рассчитываться сложные признаки с использованием специальных функций. Функции оценки решений, формирования результатов проектирования и условий проектирования формализуются и типизируются. Каждый блок ПМ является функционально автономным. Все связи (информационные и управляющие) между различными ПМ и их блоками организуются стандартным способом и могут реализовываться с помощью инвариантных программных компонентов. Пример представления алгоритма выбора решения, удовлетворяющего данному положению, представлен в [2].

6. Организация информационного взаимодействия программных комплексов и компонентов САПР ТПП, в которой можно выделить четыре уровня представления данных (рис. 3): 0 — уровень пользователя, 1 — общий уровень системы, 2 — уровень комплексов, 3 — уровень компонентов (T — трансляторы, W — программы вывода данных, E_1 и E_2 — редакторы, A — программы формирования пространства признаков, F — программы формирования результата).

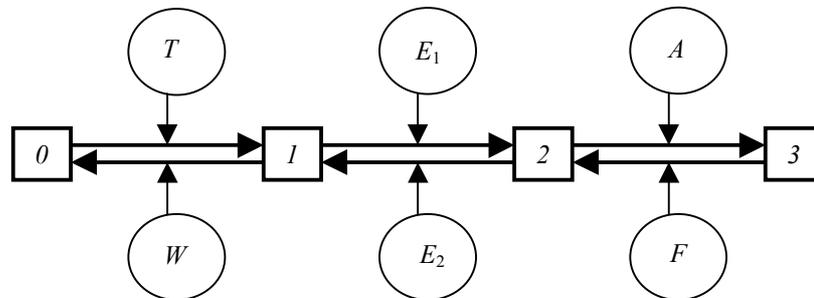


Рис. 3

Для каждого комплекса в общем случае формируется оригинальное представление заданий, а также входных и выходных данных. Это необходимо для удобства эксплуатации ПО, минимизации состава данных об объектах и процессах, возможности представления результатов в форме выходных документов.

Таким образом реализуются принципы независимости и открытости САПР ТПП, но принцип системного единства требует описания исходных данных на едином языке системы J (1-й уровень). Следовательно (рис. 4), язык J_n n -го программного комплекса (2-й уровень) системы должен являться либо составной частью языка системы J ($J_n \subseteq J$), либо между языками должно быть взаимно-однозначное соответствие ($J_n \leftrightarrow J$). То же должно быть справедливо и для выходного языка, т.е. либо язык G_n комплекса n (2-й уровень) является подмножеством выходного языка системы G (1-й уровень) ($G_n \subseteq G$), либо должно существовать между ними взаимно-однозначное соответствие ($G_n \leftrightarrow G$).

Языки J и G называются предметными языками описания полной модели объекта или процесса, а языки $\{J_n\}$ и $\{G_n\}$ — предметными языками описания частных моделей объекта или процесса. Для перехода от языка J к языкам J_n и от языков G_n к языку G описания данных необходимы редакторы (на рис. 3 — это E_1 и E_2). В этом случае модель описания объекта или процесса на языках J_n или G_n называется производной. Языки J_n и G_n должны иметь единый синтаксис, так как по мере развития системы будут добавляться новые языки, что требует открытости языков J и G .

Каждый компонент также может иметь оригинальное представление данных. Для перехода к 3-му уровню служат программы формирования пространства признаков задачи A , а для перехода от 3-го уровня — программы формирования результата F . Технология проектирования

алгоритмов и программ должна обеспечивать явное выделение пространства признаков решения задачи и правил формирования массивов результатов. Это согласуется с предложенным способом представления АМ и ПМ (см. рис. 2), где блок 1 АМ реализуется программами группы A , а блок 5 АМ — программами группы F .

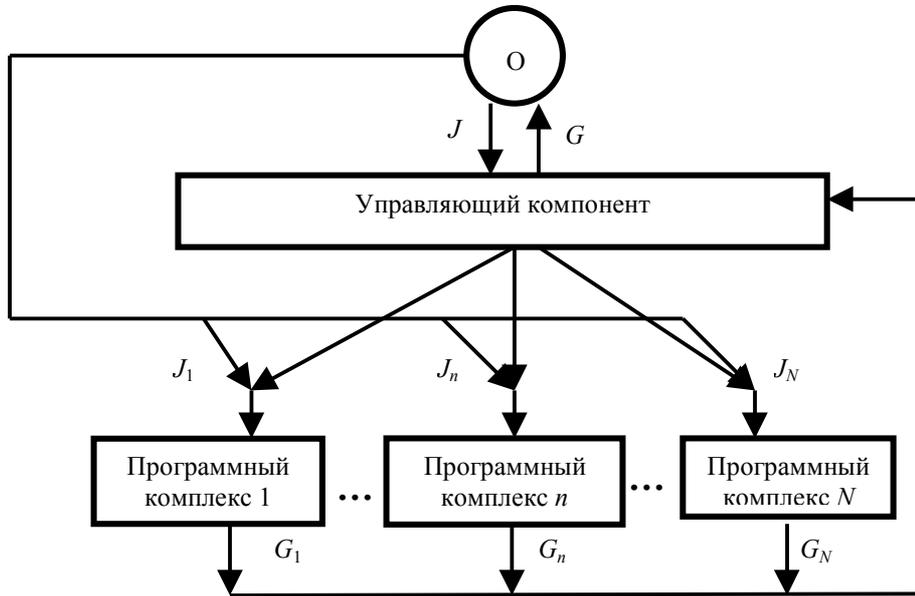


Рис. 4

7. *Декомпозиция алгоритмов.* Алгоритмы решения задач ТПП разделяются на части по правилу: „каждая часть алгоритма должна являться функцией признаков, относящихся к объектам одного класса“.

Например, решение задачи назначения инструментального перехода $P = f(D, B, I)$ зависит от параметров детали D , от возможностей оборудования B и параметров инструмента I . Можно эту задачу представить как композицию трех задач $P = P_I P_B P_D$. В задаче P_D по параметрам детали и заготовки определяются способ обработки, интервальные значения режимов резания и требования к режущему инструменту. В задаче P_B по параметрам оборудования уточняются режимы резания и требования к инструменту. В задаче P_I по требованиям к режущему инструменту определяются его параметры. Как видно, в этом случае каждая последующая задача доопределяет решение предыдущей.

Другой пример, когда при изготовлении детали необходимо минимизировать состав используемого режущего инструмента. В этом случае для каждой поверхности независимо друг от друга определяются возможные способы обработки и требования к режущему инструменту. А затем окончательное решение задачи находится пересечением всех полученных множеств возможных способов обработки.

Следовательно, каждая часть алгоритма должна „выбирать“ допустимое множество альтернативных решений, что согласуется с четвертым и пятым положениями технологии проектирования. Безусловно, такой подход усложняет проектирование алгоритмов, но при этом значительно повышается их тиражируемость.

8. *Организация инвариантных программных комплексов* для решения задач ТПП с оригинальными схемами решения и многократным использованием. Данные комплексы имеют три уровня: пакеты настройки, обработки и системы управления базами данных и знаний. Пакет обработки выполняет одну из задач ТПП и включает в свой состав лингвистические средства, ориентированные на технолога, и инвариантные программные средства, которые настраиваются на решение конкретной задачи по правилам из базы знаний. Пакет настройки предназначен для настройки пакета обработки на выполнение конкретной функции в кон-

кратных производственных условиях. В состав пакета входят средства, ориентированные на разработчика и обеспечивающие накопление баз данных и знаний. Связь между пакетами настройки и обработки осуществляется только через базы данных и знаний. Следовательно, эта связь только информационная. В работе [2] достаточно подробно представлены программные комплексы „Адрес“ и „Группа“.

Заключение. Соблюдение описанных положений, которые определяют технологию проектирования алгоритмического и программного обеспечения, позволяет генерировать САПР ТПП для конкретного предприятия и осуществлять дальнейшее ее развитие.

СПИСОК ЛИТЕРАТУРЫ

1. Митрофанов С. П., Гульнов Ю. А., Куликов Д. Д., Падун Б. С. Применение ЭВМ в технологической подготовке серийного производства. М.: Машиностроение, 1981. 287 с.
2. Митрофанов С. П., Куликов Д. Д., Миляев О. Н., Падун Б. С. Технологическая подготовка гибких производственных систем. Л.: Машиностроение, 1987. 352 с.
3. Падун Б. С. Программное обеспечение САПР ТПП. Л.: ЛИТМО, 1989. 77 с.
4. Падун Б. С. Математическое обеспечение АСТПП. Л.: ЛИТМО, 1981. 83 с.

Сведения об авторе

Борис Степанович Падун

— канд. техн. наук, доцент; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра технологии приборостроения; E-mail: bsp.tps.ifmo@mail.ru

Рекомендована кафедрой
технологии приборостроения

Поступила в редакцию
14.12.09 г.