

АКАДЕМИЯ НАУК СССР
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

УДК 519.8

ЗАДАЧИ И МЕТОДЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ В АВИАСТРОЕНИИ

Ответственный редактор
доктор физ.-матем. наук Ю.А. Флёров

В статьях сборника изучаются иерархия задач в системах автоматизированного проектирования, формирование облика летательных аппаратов, инструментальные средства САПР.

Рецензенты: Ю.Н. Павловский,
В.В. Фёдоров

Научное издание

Св. план 1989, поз. 145.

© Вычислительный центр
Академии наук СССР, 1991



Вычислительный центр АН СССР
Москва 1991

ИЕРАХИЯ ЗАДАЧ ПРОЕКТИРОВАНИЯ

П.С. Краснощёков, Ю.А. Флёров

Постановка задачи проектирования как задачи синтеза системы с заданными качествами представляет собой сложный многоэтапный процесс и осуществляется на стадии внешнего проектирования во взаимодействии с представителями внутреннего проектирования. Можно выделить следующие основные этапы, которые проходят постановка задачи проектирования.

На первом этапе определяются цели проектирования. Цели внутреннего проектирования формулируются на этапе внешнего проектирования.

На втором этапе происходит определение объекта проектирования, формирование концепции проектируемой системы, подготавливаются данные, необходимые для синтеза математической модели проектируемого объекта.

Синтез математической модели объекта проектирования включает: разработку математического описания собственно технической системы в виде набора проектных параметров и ограничений, составляющих структурно-параметрическую модель объекта проектирования, и модели функционирования проектируемой системы.

На следующих этапах происходит формализация задачи проектирования, т.е. появляется её строгая математическая формулировка, пригодная для использования ЭВМ. На этих этапах формируется область поиска решения и формализуются система предпочтений и принцип оптимальности. Особенно сложен этап формализации системы предпочтений и принципа оптимальности. Дело в том, что, используя формальные конструкции, удаётся добиться лишь приближения к тем принципам оптимальности, которыми оперирует заказчик при оценке системы. Как правило, представления заказчика

об оптимальности богаче по своему содержанию, чем те, которые он сообщил в техническом задании (ТЗ) и которые удалось выявить и формализовать.

Перейдём к формальному определению задачи проектирования. Предположим, удалось определить и формализовать область поиска решения, т.е. исходное множество альтернатив X , среди которых мы надеемся отыскать хотя бы одну альтернативу, удовлетворяющую нашему представлению об оптимальности. Для этого нужно уметь формализовать ту систему предпочтений, носителем которой является заказчик.

Обычно представители внешнего проектирования сами тем или иным способом генерируют рабочее множество альтернатив для формирования системы предпочтений и принципа оптимальности. Эти альтернативы сравниваются между собой и с аналогичными зарубежными образцами в некоторых предполагаемых условиях функционирования на определённых типичных режимах эксплуатации с привлечением имитационных экспериментов, на основе экспертиз и прочих присущих этой стадии процедур. При этом имитируются весьма сложные условия и режимы функционирования будущей системы, поэтому сравнению может быть подвергнуто лишь небольшое конечное число альтернатив. Этого, конечно, оказывается недостаточно, чтобы построить адекватную аппроксимацию предпочтений некоторой формальной моделью выбора (X, R) , состоящей из множества X альтернатив и бинарного отношения R на нём. Итак, первый вывод: система предпочтений заказчика может быть реализована на относительно небольшом конечном множестве альтернатив $X^* \subset X$, поэтому модель (X^*, R) не может служить моделью выбора на стадии внутреннего проектирования, так как для этого необходимо иметь систему предпочтений на всём исходном множестве альтернатив X . В силу этих обстоятельств традиционной формой технического задания является список требований к функционально-техническим характеристикам будущей системы.

В этих требованиях как-то отражена та система предпочтений, которая вырабатывается на стадии внешнего проектирования, причём негласно предполагается, что эти требования выделяют будущую систему из множества возможных. К сожалению, в реальности дело обстоит не так благополучно, как это может показаться. Дело в том, что требования ТЗ сформулированы очень жёстко, что зачастую не позволяет конструктору найти компромисс между этими требованиями и возможностями конструкторского бюро (КБ). Откровенно говоря, неясна процедура формирования этих требований. Нередко бывает так, что их определяют, отталкиваясь от какого-либо известного образца прототипа, простым увеличением значений тех характеристик, которые желательно увеличить, и уменьшением значений тех, которые желательно уменьшать. Такой voluntarizm приводит к тому, что требования подобного ТЗ не удаётся реализовать. Таким образом, мы приходим ко второму выводу: традиционная форма, в которую облечены требования технического задания, некорректна.

Выходом из положения может быть лишь организация грамотного, научно обоснованного информационного взаимодействия между стадиями внешнего и внутреннего проектирования, основанного на естественном разделении сфер влияния при постановке задачи проектирования. Нам кажется разумной следующая схема. Во-первых, генерация рабочего множества альтернатив X^* будущей системы (на котором представители внешнего проектирования формируют модель выбора (X, R) , т.е. свою систему предпочтений) должна осуществляться представителями внутреннего проектирования, так как именно они по роду своей деятельности обладают необходимым профессионализмом для решения этой задачи. Они больше, чем кто-либо другой, знают возможности современной инженерной технологии и обладают опытом реального проектирования. Во-вторых, задание на формирование этого множества

альтернатив должно исходить от представителей внешнего проектирования, так как именно они лучше всего представляют себе цели, ради которых создаётся система. Если признать такую схему естественной и разумной, то очевидно, что необходимо разработать новую процедуру формирования технического задания, адекватную этой схеме. Эта процедура представляет собой выделение главных этапов обмена информацией между представителями внешнего и внутреннего проектирования и постановки конкретных задач, которые необходимо решать на этих этапах.

Так как в обсуждаемой схеме предлагается рабочее множество альтернатив X^* формировать для представителей внешнего проектирования на стадии внутреннего проектирования, то должна быть определена система предпочтений – модель выбора (X, Φ) , которую необходимо сообщить представителям внутреннего проектирования. Эта модель выбора должна соответствовать информационному уровню внутреннего проектирования, т.е. реализации отношения должна быть проверяема на этом уровне. Если такое отношение удалось определить, то задача проектирования на первом этапе формулируется следующим образом: выделить множество $X^* \subseteq X$ максимальных элементов в модели (X, Φ) :

$$X^* = \text{Max}(X, \Phi).$$

При удачном выборе модели (X, Φ) мы вправе ожидать, что анализ проектов из множества X^* позволит сделать выбор, удовлетворяющий конечным целям внешнего проектирования.

Модель (X, Φ) выбрана удачно, если отношение Φ в определённом смысле универсально, т.е. инвариантно к совокупности требований, которые могут быть сформулированы на стадии внешнего проектирования. Это означает, что среди множества альтернатив $X^* = \text{Max}(X, \Phi)$ всегда найдётся проект $x \in X$, удовлетворяющий требованиям заказчика системы. В таком случае мы будем говорить, что отношение Φ согласовано с целями внешнего проектирования.

Если Φ окажется таково, что $\text{Max}(X, \Phi) = X$, то пользы от такой модели выбора, очевидно, не будет. Поэтому необходима "золотая середина". Далее будет указан один из способов определения такой "золотой середины" для динамически управляемых систем. Анализ показывает, что требования, выраженные в ТЗ, почти всегда сводятся к списку характеристик проектируемой системы, значения которых определены тем или иным способом. Перечисляются также набор стандартных ситуаций, соответствующий определённым формам эксплуатации системы. Формально все эти требования выливаются в задание некоторой совокупности систем равенств и неравенств, которым подчинены характеристики проектируемой системы.

Сами характеристики, как правило, могут быть выражены в виде функций, зависящих лишь от конструктивных параметров системы: $U(x) = \{U_1(x), U_2(x), \dots, U_m(x)\}$. Эти характеристики сформировались в результате длительного опыта проектирования и эксплуатации систем аналогичного назначения и выражают главные качества, определяющие совершенство этих систем. Поэтому вполне естественной является мысль использовать их для формирования отношения предпочтения Φ в модели (X, Φ) . Для этого из набора $U(x) = \{U_1(x), U_2(x), \dots, U_m(x)\}$ необходимо выделить характеристики $W(x) = \{W_1(x), W_2(x), \dots, W_n(x)\}$, "ответственные" за эффективность функционирования будущей системы. Условимся $W(x)$ называть вектором частных критериев эффективности. Компоненты этого вектора $W_i(x) = 1, 2, \dots, n$, отличаются от остальных характеристик тем свойством, что по смыслу целевого назначения системы их выгодно увеличивать.

Остальные характеристики $V(x) = \{V_1(x), V_2(x), \dots, V_m(x)\}$ используются при формировании ограничений, входящих в состав условий, выделяющих множество X . Ранее мы условились считать, что множество X так или иначе уже определено, поэтому сосредоточим пока внимание на характеристиках $W(x)$ и определении с их помощью отношения Φ .

Посредством векторного критерия эффективности $W(x)$ можно задать три типа бинарных отношений:

$x\Phi_1 y$ тогда и только тогда, когда $W_i(x) \geq W_i(y), i = 1, 2, \dots, n$;

$x\Phi_2 y$ тогда и только тогда, когда $W_i(x) > W_i(y), i = 1, 2, \dots, n$;

$x\Phi_3 y$ тогда и только тогда, когда $W_i(x) \geq W_i(y), i = 1, 2, \dots, n$; причём хотя бы для одного i имеет место строгое неравенство, т.е. $W(x) \neq W(y)$.

Отношение Φ_2 называют отношением строгого доминирования, или *отношением Слейтера*, а отношение Φ_3 — *отношением Парето*. Множество максимальных элементов в X , соответствующих этим отношениям, т.е. $\text{Max}(x, \Phi_2)$ и $\text{Max}(x, \Phi_3)$, обозначим через $S(X, W)$ и $P(X, W)$.

Множество $P(X, W)$ называют множеством эффективных (оптимальных по Парето) альтернатив, а $S(X, W)$ — множеством полужэффективных (слабоэффективных, оптимальных по Слейтеру) альтернатив.

Вернёмся к задаче проектирования. Итак, на первом этапе решения задачи проектирования предлагается отказаться от "жесткой" формы требований ТЗ преобразовать её в более "слабую" в виде модели выбора (X, Φ_3) или даже (X, Φ_2) . Задачей внутреннего проектирования (на этапе формирования облика будет являться построение множеств альтернатив $P(X, W)$ или $S(X, W)$ и передача их для дальнейшего анализа на стадию внешнего проектирования, на которой эти множества должны быть использованы для уточнения систем предпочтений и принципа оптимальности, т.е. формирования модели (P, R) и (S, R) , и решения задачи выделения из них максимальных элементов $X_{\Phi_3, R} = \text{max}(P, R)$

или $X_{\Phi_2, R} = \text{max}(S, R)$. Альтернативы $X_{\Phi_3, R}$ или $X_{\Phi_2, R}$ должны лечь основу уточнённого ТЗ, форма которого уже может быть традиционной, поскольку она будет лишена упомянутых ранее недостатков. Фактически на стадию внутреннего проектирования будет передано описание одной и нескольких альтернатив из множеств P или S , а так как их построение было

выполнено представителями внутреннего проектирования, то следует ожидать, что былых противоречий между требованиями ТЗ и возможностями внутреннего проектирования не возникнет.

Теперь можно представить, как должно быть организовано информационное взаимодействие между стадиями внешнего и внутреннего формирования облика.

1. От представителей внешнего проектирования поступает набор характеристик $V = \{V_1, V_2, \dots, V_m\}$, подчинённых необходимым требованиям, и вектор частных критериев эффективности $W = \{W_1, W_2, \dots, W_n\}$, на основе которого сформулирована система предпочтений Φ .

2. Представители внутреннего проектирования формируют с учётом упомянутых требований исходное множество: альтернатив будущей системы X . Таким образом, у представителей внутреннего проектирования появляется модель выбора (X, Φ) .

3. Представители внутреннего проектирования решают задачу первого этапа синтеза облика будущей системы – определяют множество X_Φ максимальных элементов в модели (X, Φ) и сообщают его представителям внешнего проектирования.

4. На стадии внешнего проектирования решают задачу второго этапа синтеза облика системы – определяют на основе анализа элементов из множества X_Φ окончательную систему предпочтений R и строят множество максимальных элементов $X_{\Phi R}$ в модели (X_Φ, R) .

5. Если среди элементов множества $X_{\Phi R}$ найдутся варианты системы, соответствующие представлениям заказчика об оптимальности, то их описание ложится в основу окончательного варианта ТЗ, которое поступает на стадию внутреннего проектирования для формирования технического предложения на детальную разработку проекта.

Если среди вариантов $X_{\Phi R}$ не нашлось удовлетворительных (или представителей внешнего проектирования не удалось на основе анализа множества X_Φ окончательно уточнить систему предпочтений R), то исходные требования и система предпочтений Φ должны быть пересмотрены. Это означает, что должны быть пересмотрены требования к характеристикам V_1, V_2, \dots, V_m и вектору частных критериев эффективности W_1, W_2, \dots, W_n . После формирования новых требований и нового вектора частных критериев вышеописанная процедура должна быть повторена. Если разумное число таких итераций не даёт удовлетворительных результатов, то необходимо либо снизить требования, либо избреть принципиально новую систему, кардинально меняя структуру множества X .

Описанная итеративная процедура обмена информацией между представителями внешнего и внутреннего проектирования и представляет собой содержательную сторону диалога "заказчик — проектировщик".

Чтобы осуществить предложенную процедуру постановки и решения задач проектирования, необходимо разработать соответствующее ей математическое обеспечение. На этом пути можно выделить следующие основные препятствия, без преодоления которых надеяться на успех не приходится. Перечислим их том порядке, в котором они естественно возникают при первых же попытках решать "в лоб" указанную проблему:

- структура множества X (в особенности для сложных технических систем чрезвычайно сложна, а размерность вектора проектных параметров $x \in X$ очень велика;
- отношение предпочтения Φ трудно проверяемо, т.е. компоненты вектора частных критериев эффективности $W(x)$ являются сложными трудновычислимыми функциями;

- представление и оценка компонент вектора $W(x)$, т.е. проверка реализации отношения Φ сложна и, как правило, представляет собой совокупность разнородных взаимосвязанных моделей;
- весьма непростая проблема построения "универсального" отношения Φ , так как качество функционирования системы зависит не только от её конструктивного облика, но также от качества управления ею и внешних условий функционирования.

По крайней мере, первые три из перечисленных сложностей завязаны в один Gordiev узел, и разрубить его можно, лишь осуществив дальнейшую декомпозицию задачи проектирования. Это приведёт к построению иерархической структуры задач внутреннего проектирования, один из возможных вариантов которой мы попытаемся здесь схематично описать. Проблему построения "универсального" отношения Φ мы обсудим позднее, а пока для простоты изложения будем считать, что мы умеем её решать.

Для того чтобы бороться со сложностью описания проектируемой системы, т.е. со сложностью структуры множества X и большой размерностью вектора x , естественно попытаться ввести иерархию уровней детализации описания конструкции системы [1].

Пусть набор проектных параметров $x \in X \subset D^N$, где D^N – декартово пространство размерности N . Это означает, что выбор конкретного значения $x \in X$ с определённой степенью детализации определяет конструкцию системы, т.е. можно создать систему с вектором проектных параметров x . Отметим, что x может давать далеко не самое подробное описание системы. Основное требование состоит в том, чтобы задание $x \in X$ не вызывало сомнений в реализации проекта системы с параметрами x .

Как мы помним, первым этапом решения задачи проектирования являлось выделение максимальных элементов в модели (X, Φ) . Эту процедуру

для краткости мы будем в дальнейшем обозначать следующим образом:

$$X \rightarrow X_\Phi = \max(X, \Phi). \quad (1)$$

Предположим, что на принятом уровне детализации описания системы решение задачи (1) представляет непреодолимые трудности. Введём тогда несколько уровней описания, причём описание системы на соседних уровнях свяжем между собой соотношениями:

$$\begin{aligned} x^{k+1} &= f_{k+1}(x^k), \quad X^{k+1} = f_{k+1}(X^k), \\ x^k &\in X^k \subset D^N, \quad x^{k+1} \in X^{k+1} \subset D^{N_{k+1}}, \end{aligned}$$

где $k = 0, 1, \dots, m-1$, m — число уровней, $N_{k+1} \leq N_k$, f_{k+1} — отображение, которое переводит множество X^k в множество X^{k+1} .

Каждый вектор x^{k+1} получается из x^k агрегированием при помощи вектор-функции f_{k+1} и даёт при $N_{k+1} \leq N_k$ более целостное описание системы, чем x^k . На каждом $k+1$ -м уровне задаётся отношение Φ_{k+1} , причём предполагается, что определение множества максимальных элементов $X_{\Phi_{k+1}}$ в модели (X^{k+1}, Φ_{k+1}) представляет собой менее трудоёмкую задачу, чем аналогичного множества X_{Φ_k} в модели (X^k, Φ_k) .

Агрегирование — очень действенный приём, которым часто пользуются на практике. Формы агрегирования весьма разнообразны и существенно зависят от содержательного смысла решаемых задач. Так, в самолётостроении вектор x^k большой размерности может описывать крыло очень сложной формы в плане, а x^{k+1} — эквивалентное ему по основным аэродинамическим характеристикам трапециевидное крыло. Отношения Φ_{k+1} и Φ_k могут иметь один и тот же содержательный смысл, но будут различаться по полноте учёта конструкторской информации. Отношение Φ_{k+1} строится на более общей информации о системе, чем Φ_k , поэтому менее чувствительно к деталям описания. В отношении Φ_k обычно учтены дополнительные (по сравнению с Φ_{k+1}) требования к

системе, естественно возникающие при детализации описания. Каждому уровню описания соответствует определённая форма модели функционирования. В авиастроении, например, широко используются следующие при уровня описания движения самолёта. На самом "грубом" уровне самолёт представляется как материальную точку (три степени свободы), и его движение описывается соответствующими уравнениями движения центра масс. Следующий уровень детализации представляет самолёт как твёрдое тело (шесть степеней свободы). К уравнению движения центра масс добавляются уравнения моментов. Моментами управляют с помощью отклоняемых поверхностей, поэтому модель движения дополняется моделью функционирования элементов управления. На третьем уровне учитывается, что самолёт является упругим телом (число степеней свободы неограниченно возрастает), и его уравнения движения ещё более усложняются. На каждом уровне описания число требований к самолёту увеличивается, следовательно, усложняется система отношений предпочтения Ф. Для каждого класса проектируемых систем содержательная сторона процедуры выделения уровней агрегирования имеет свою специфику и во многих отношениях уникальна. Это подтверждает мысль о том, что невозможно создать универсальную САПР, инвариантную к различным объектам проектирования. Однако основные принципы, положенные в основу построения систем автоматизированного проектирования, по-видимому, едины. О них мы и попытаемся говорить в этой статье.

Продолжим наши построения. Наряду с функциями f_{k+1} нас будут интересовать обратные функции f_{k+1}^{-1} , восстанавливающие по заданному x^{k+1} его преобразы x^k . Не следует думать, что f_{k+1} и f_{k+1}^{-1} задаются аналитически. Напротив, чаще всего процедура их вычисления на ЭВМ сложна и трудоёмка.

Теперь, отпавляясь от введённых функций агрегирования и отношений Φ_{k+1} , рассмотрим следующее рекуррентное соотношение:

$$X_{\Phi_k}^* = \max(f_{k+1}^{-1}(\Phi_{k+1}^*), \Phi_k), \quad (2)$$

$$k = m-1, m-2, \dots, 1, 0$$

с начальным условием $X_{\Phi_m}^* = X_{\Phi_m} = \max(X^m, \Phi_m)$. Здесь $f_{k+1}^{-1}(\Phi_{k+1}^*)$ — новый прообраз множества $X_{\Phi_{k+1}}^*$ при отображении $f_{k+1}: X^k \rightarrow X^{k+1}$.

В качестве решения задачи проектирования примем множество $X_{\Phi_0}^*$.

Процесс нахождения $X_{\Phi_0}^*$ из (2) удобно записывать в виде диаграммы

$$X_{\Phi_{k+1}}^* \rightarrow X_{\Phi_k}^* = \max(f_{k+1}^{-1}(\Phi_{k+1}^*), \Phi_k). \quad (3)$$

Сама последовательная процедура нахождения элементов множеств $X_{\Phi_k}^*$ представляет собой детализацию вариантов проекта при переходе от одного уровня описания к другому, т.е. процедуру синтеза альтернатив будущей системы. Следует ожидать, что при "хороших" свойствах f_k и Φ_k построение множества $X_{\Phi_0}^*$ в отличие от прямого ("в лоб") отыскания множества $X_{\Phi_0} = \max(X^0, \Phi_0)$ может быть практически реализовано современными математическими методами на ЭВМ. Однако здесь возникает весьма непростая проблема: насколько множество $X_{\Phi_0}^*$, которое мы условились принимать за решение задачи проектирования, приемлемо в качестве представления множества X_{Φ_0} ?

Ясно, что проблема может считаться решённой удовлетворительно, если удастся гарантировать выполнение какого-либо из следующих условий:

$$а) X_{\Phi_0} = X_{\Phi_0}^*, \quad б) X_{\Phi_0} \subseteq X_{\Phi_0}^*$$

или

$$в) (X_{\Phi_0} = X_{\Phi_0}^*)/I_{\Phi_0}, \quad г) (X_{\Phi_0} \subseteq X_{\Phi_0}^*)/I_{\Phi_0}$$

в случае транзитивности отношения Φ_0 . Здесь запись вида $(Z \subseteq Y)/I_{\Phi}$ означает включение Z в Y с точностью до отношения эквивалентности I_{Φ} , порождённого отношением Φ , а $(Z = Y)/I_{\Phi}$ — совпадение множеств Z и Y

с точностью до отношения эквивалентности I_Φ . Требование включения X_{Φ_0} в $X_{\Phi_0}^*$ или их совпадения с точностью до отношения эквивалентности, порождающего отношение Φ_0 , естественно, поскольку очевидно, что при этом не теряется ни одного "оптимального" варианта проектируемой системы.

В общем случае множество $X_{\Phi_0}^*$, найденное из (3), может не содержать ни одного "оптимального" варианта из множества X_{Φ_0} . Поэтому для корректности перехода от задачи $X^0 \rightarrow X_{\Phi_0} = \max(X^0, \Phi_0)$ к задаче (3), т.е. для выполнения одного из условий а) – г), отношения Φ_k и агрегирование должны быть определённым образом согласованы [1]. На содержательном уровне согласованность Φ_k с Φ_{k+1} означает следующее. Пусть на $k+1$ -м уровне иерархии процесса проектирования альтернатива x предпочтительнее y по отношению Φ_{k+1} . Тогда при переходе на более детальный уровень описания системы может быть сохранена частичная упорядоченность альтернатив по отношению Φ_k . Если сопоставить каждому уровню конструктора, задающего отношение предпочтения, то сказанное означает, что конструкторы соседних уровней иерархии одинаково сравнивают проекты, одинаково понимают, что такое "лучшая система".

Легко видеть, что m -й уровень агрегирования, дающий наиболее целостное описание системы, и есть тот самый уровень, который должен быть принят на этапе формирования облика. Так как система формирования облика связывает стадии внешнего и внутреннего проектирования, то её ошибки сродни ошибкам генерального конструктора. Эти ошибки, как правило, не могут быть исправлены на других ступенях стадии внутреннего проектирования.

Вернёмся к диаграмме (3). Из неё видно, что процедура нахождения множеств $X_{\Phi_k}^*$ состоит из последовательных шагов, на каждом из которых решаются два типа задач: построение обратного отображения f_{k+1}^{-1} , определяющего

полный прообраз множества $X_{\Phi_{k+1}}^*$, выделенного на предыдущем шаге, и решение задачи построения множества максимальных элементов в модели $(f_{k+1}^{-1}(X_{\Phi_{k+1}}^*), \Phi_k)$. Первая задача относится к классу так называемых обратных задач, а вторая — к классу задач многокритериальной оптимизации. Таким образом, процесс синтеза альтернатив будущей системы представляет собой последовательную цепочку решения обратных задач и задач оптимизации.

Итак, пусть со стадии внешнего проектирования на стадию внутреннего проектирования поступили отношение предпочтения Φ_m , и некоторая совокупность требований к характеристикам системы. На стадии внутреннего проектирования (этап формирования облика) сформулировали исходное множество альтернатив X^m и модель выбора (X^m, Φ_m) . На основе этой модели на этапе формирования облика построили "рабочее" множество альтернатив $X_{\Phi_m}^* = \max(X^m, \Phi_m)$ для представителей внешнего проектирования, которые в результате анализа этого множества сформировали корректное ТЗ и передали его представителям внутреннего проектирования. Это ТЗ выделено из множества $X_{\Phi_m}^*$ одну или несколько альтернатив (для простоты изложения будем считать, что одну, и обозначим её x_0^m), которые легли в основу разработки технического предложения на проектирование системы. Началась процедура детализации проекта: строятся все прообразы $f_m^{-1}(x_0^m)$ проекта x_0^m и поступают в качестве альтернатив технического предложения на уровень $m-1$; на этом уровне из них выделяются максимальные элементы по отношению Φ_{m-1} — все они (если агрегирование было выполнено правильно) удовлетворяют требованиям, предъявляемым к проекту x_0^m ; из них выделяется проект x_0^{m-1} (если нет дополнительного предпочтения, то выбирается любой), и строятся все его прообразы с помощью отображения f_{m-1}^{-1} ; они передаются на уровень $m-2$ и т.д., пока эта процедура не приведёт к выделению проекта x_0^0 на самом детальном, "нулевом" уровне. Образуется цепочка описаний системы

(проектов) $x_0^m \rightarrow x_0^{m-1} \rightarrow \dots \rightarrow x_0^1 \rightarrow x_0^0$, дающая полное описание будущей системы на всех уровнях детализации. Такова схема процедуры синтеза проекта системы. Как уже говорилось, в её основе лежит последовательное решение обратных задач и задач оптимизации. Это очень трудные задачи. Сложность проблемы синтеза во многом является следствием трудности решения указанных задач. Они, как правило, некорректны и требуют для своего решения изощрённых математических методов, разработкой так называемых быстрых алгоритмов и по возможности "упрощённых" моделей функционирования. Предложенная схема, как и всякая схема, является идеализацией. В реальности всегда будут возникать отклонения, обусловленные неполнотой математического обеспечения, несогласованностью предпочтений, неточностью моделей функционирования и т.п. Поэтому каждое описание системы в цепочке $x_0^m \rightarrow x_0^{m-1} \rightarrow \dots \rightarrow x_0^1 \rightarrow x_0^0$ должно быть подвергнуто анализу на максимум точных и полных моделях (насколько это позволяют современные достижения в каждой области знания) с привлечением точных алгоритмов расчёта характеристик и режимов работы системы, полученных в экспериментальных исследованиях на натуральных и полунатурных моделях, и т.д. Здесь не стоит столь остро (как в задаче синтеза) проблема быстроты выполнения расчётов, поскольку решаются прямые задачи, которые, как известно, по вычислительной сложности уступают обратным задачам и задачам оптимизации. Таким образом, математическое обеспечение процесса решения задачи проектирования по своему назначению подразделяются на две большие системы: систему анализа и систему синтеза. Систему анализа составляют точные и полные модели и алгоритмы поверочного расчёта, систему синтеза – упрощённые модели и быстрые алгоритмы решения обратных и оптимизационных задач. Система синтеза должна непрерывно совершенствоваться, пополняться новыми моделями и алгоритмами. С ростом быстродействия и объёма памяти ЭВМ многие модели системы анализа будут переходить в систему синтеза, а система

анализа будет пополняться ещё более совершенными и точными моделями и алгоритмами. Взаимодействие системы синтеза и системы анализа имеет интегральный характер. Каждый выполненный синтез проверяется и уточняется на системе анализа, и эта процедура повторяется до тех пор, пока не будет получен проект, удовлетворяющий проектировщика и заказчика.

Идея построения иерархии задач проектирования может быть использована также и на стадии внешнего проектирования для построения последовательной процедуры отбраковки альтернатив, основанной на использовании так называемого метода последовательного анализа вариантов. Принципиальная схема последовательного анализа вариантов аналогична схеме (3). Строится последовательность согласованных отношений $R_j, j = 1, 2, \dots, k$, и отбраковка вариантов происходит по схеме

$$X^{j-1} \rightarrow X^j = \max(X^{j-1}, R_j).$$

Для построения последовательности согласованных отношений R_j необходимо иметь иерархию моделей функционирования верхнего уровня, отражающих конкретную обстановку, в которой придётся функционировать будущей системе. Для динамических систем типа летательных аппаратов на одной из ступеней указанной иерархии моделей верхнего уровня можно использовать модели позиционных дифференциальных игр «сближения – уклонения».¹

На этом мы закончим обсуждение проблемы дальнейшей отбраковки альтернатив и вернёмся к упомянутому выше "универсальному" отношению. Однако в наиболее простых случаях можно построить ему эквивалентное, сформировав вектор частных критериев эффективности. Примером построения такого отношения мы и закончим.

¹ Красовский Н.И., Субботин А.И. Позиционные дифференциальные игры. М.: Наука, 1974.

Рассмотрим упрощённую модель движения самолёта, которая обычно используется на этапе формирования облика. Имеем¹

$$\begin{aligned} \dot{x} &= V \cos \theta \cos \psi, & \dot{H} &= V \sin \theta, \\ \dot{z} &= -V \cos \theta \sin \psi, \\ \dot{V} &= g \left(\frac{P}{G} - \frac{C_{x_0} q}{G/S} - \frac{BG/S}{q} n_y^2 - \sin \theta \right), \end{aligned} \quad (4)$$

$$\begin{aligned} V \dot{\theta} &= g (n_y \cos \gamma - \cos \theta), \\ V \dot{\psi} \cos \theta &= -g n_y \sin \gamma. \end{aligned}$$

Здесь первые три уравнения определяют положение центра масс самолёта в земной системе координат (H – высота полёта, V – величина скорости; θ и ψ – углы наклона траектории и курса; γ – угол крена, G – вес самолёта; P – тяга двигателя; g – ускорение свободного падения; q – скоростной напор; S – площадь крыла; C_{x_0} , B – аэродинамические коэффициенты; n_y – перегрузка).

Фазовыми переменными в (4) являются x , H , z , V , θ , ψ , а управлениями – n_y , γ , P . К проектным параметрам относятся геометрические параметры крыла, фюзеляжа и оперения, допустимая температура газов перед турбиной двигателя, материал конструкции и др. Они вполне определяют правые части уравнений (4), так как позволяют по соответствующим математическим моделям расчитать тягу двигателя, аэродинамические характеристики и вес самолёта.

На фазовые переменные и управления наложены ограничения вида:

$$\begin{aligned} H &\geq 0, & \frac{\rho(H)V^2}{2} &< q_{\max}, & \frac{V}{a(H)} &\leq M_{\max}, \\ -\pi &\leq \gamma \leq \pi, & 0 &\leq n_y \leq n_{y, \max}, & 0 &\leq P \leq P_{\max}, \end{aligned}$$

где $\rho(H)$ – плотность воздуха, $a(H)$ – скорость звука, q_{\max} , M_{\max} , n_{\max} – макси-

¹ *Острославский И.В., Стражева И.В.* Динамика полёта. Траектории летательных аппаратов. М.: Машиностроение, 1969.

мально допустимые значения скоростного напора, числа Маха и перегрузки.

В этом примере [2] удаётся построить выпуклую оболочку правых частей системы (4). В пространстве с осями

$$f_1 = \frac{V}{g} + \sin \theta, \quad f_2 = \frac{\theta V}{g} + \cos \theta, \quad f_3 = \frac{\psi V \cos \theta}{g}$$

эта оболочка имеет форму тела вращения, напоминающего пулю. Её граница образована параболической поверхностью

$$f_2^2 + f_3^2 + (f_1 - W_4)W_3 = 0, \quad (5)$$

цилиндрической

$$f_2^2 + f_3^2 - W_1^2 = 0 \quad (6)$$

и плоскостью

$$f_1 = -W_2, \quad (7)$$

где

$$\begin{aligned} W_1 &= n_{y, \max}, & W_2 &= \frac{C_{x_0} q}{G/S} + \frac{BG/S}{q} n_{y, \max}^2, & W_3 &= \frac{q}{BG/S}, \\ W_4 &= \frac{P_{\max}}{G} - \frac{C_{x_0} q}{G/S}. \end{aligned} \quad (8)$$

Из (5) – (8) непосредственно видно, что с увеличением W_i , $i = 1, \dots, 4$, размеры "пули" также увеличиваются. Таким образом, величины W_i , определённые в (8), могут быть приняты в качестве критериев (функционально-технических характеристик) самолёта, задающих отношение Парето или отношение Слейтера. Заметим, что (8) – минимальная система частных критериев, характеризующая динамические возможности самолёта. В самом деле, требуются три параметра для определения размеров "пули" и ещё один параметр для фиксации её положения на оси f_1 . Критерии (8) имеют прозрачный инженерный смысл: W_1 – нормальная перегрузка – характеристика разворота; W_2 – минимальная продольная перегрузка – характеристика торможения;

W_3 — характеристика несущей способности крыла; W_4 — максимальная продольная перегрузка — характеристика разгона.

Иерархический подход к процессу проектирования, изложенный выше, нуждается в развитии. Дело в том, что любая сложная техническая система при достаточно подробном описании распадается на подсистемы, дальнейшее проектирование каждой из которых ведётся в какой-то степени независимо от других. Так, например, самолёт на определённой стадии проектирования может рассматриваться как система, состоящая из подсистем-агрегатов: крыла, фюзеляжа, двигателя и т.д. Именно блочная структура сложных систем обеспечивает успех их практического проектирования. Ясно также, что это обстоятельство во нельзя не учитывать при создании систем автоматизированного проектирования, при распараллеливании процесса автоматизированного проектирования.

Описать иерархический процесс проектирования технических систем с учётом их блочной структуры можно на основе функций агрегирования.

Рассмотрим следующие две схемы декомпозиции задачи проектирования:

"параллельную" схему

$$\begin{aligned} X_j &\rightarrow A_j = \max(X_j, \Phi_j); \quad j = 1, \dots, s; \\ A &= \bigcup_{j=1}^s d_j^{-1}(A_j) \rightarrow A_0 = \max(A, \Phi); \\ X_j &= d_j(X), \end{aligned} \quad (9)$$

"последовательную" схему

$$\begin{aligned} B_{j+1} &\rightarrow B_j = \max(d_j \circ d_{j+1}^{-1}(B_{j+1}), \Phi), \\ j &= s-1, \dots, 0, \end{aligned} \quad (10)$$

где $B_s = \max(X_s, \Phi_s)$, $\Phi_0 = \Phi$, $d_0 : 2^X \rightarrow 2^X$ — тождественное преобразование, $d_j \circ d_{j+1}^{-1}$ — композиция отображений $d_j(\circ)$ и $d_{j+1}^{-1}(\circ)$, $X_j = d_j(X)$.

Приведём содержательную трактовку схем (9), (10).

Пусть проектируемая система S состоит из подсистем s_j , $j = 1, 2, \dots, s$. Варианты подсистемы s_j описываются вектором конструктивных параметров $x_j = d_j(x)$. В простейшем случае $x = (x_1, \dots, x_s)$; в общем случае один и те же конструктивные параметры могут участвовать в описании различных подсистем, т.е. подсистемы могут "пересекаться". Отношения Φ и F_j задаются через свёртки технических характеристик системы и подсистем соответственно. Схема (9) отвечает параллельному, (10) — последовательному проектированию подсистем.

Например, базовый ЛА состоит из планёра и силовой установки. Планёр, в свою очередь, включает в себя крыло, фюзеляж, оперение, шасси и т.д. Схема (9) означает параллельное проектирование силовой установки и планёра на нижнем уровне иерархии и ЛА — на верхнем. Схема (10) — это проектирование силовой установки на нижнем и ЛА в целом — на верхнем уровне.

Говоря о практической реализации таких схем автоматизированного проектирования на ЭВМ, уже сейчас нельзя не отметить сложные проблемы, связанные с описанием объекта проектирования, с передачей информации САПР, диалогом и т.д. В особенности это относится к параллельным схемам проектирования. Они вполне соответствуют сложившейся практике проектирования подсистем сложной технической системы в КБ и, вероятно, окажутся полезными при создании САПР на многомашинной основе.

Здесь необходимо отметить следующее. Выбор параметров связей между подсистемами выходит за рамки проблемы проектирования подсистем и относится ко всей системе в целом. В САПР эти вопросы должны решаться специальной управляющей программой, роль которой аналогична роли главного конструктора в КБ: координировать действия подсистем и решать оптимизационные задачи в схеме (9). Таким образом, управление САПР технических систем с блочной структурой не сводится только к передаче информации между САПР подсистем. В процессе управления должно

происходить оперативное решение задач оптимизации, т.е. получение новой информации, которая отсутствует на уровне подсистем. Далее, при создании САПР системное математическое обеспечение должно учитывать специфику проектируемого объекта, в противном случае оно может оказаться слишком громоздким и неэффективным. Отсюда следует, что к созданию инвариантного математического обеспечения САПР нужно подходить осторожно. Разработчики САПР для успешной работы обязаны изучить объект проектирования. Когда речь идёт о действительно сложных технических системах, такой подход, как правило, себя оправдывает и позволяет избежать лишних затрат.

Если рассматривать понятие "проектирование" предельно широко, то САПР, по существу, и есть новая информационная технология научно-технического и социального прогресса общества. Совершенство систем автоматизированного проектирования будет в недалёком будущем определять интеллектуальное и техническое могущество государства. При этом нужно помнить, что совершенство это само по себе не возникнет, если мы не вложим в него труд, знания и энергию.

Литература

1. Краснощёков П.С., Морозов В.В., Фёдоров В.В. Последовательное агрегирование в задачах внутреннего проектирования технических систем // Изв. АН СССР. Сер. Техн. киберн. 1979. № 5. С. 5-12.
2. Краснощёков П.С., Петров А.А., Фёдоров В.В. Информатика и проектирование // Новое в жизни, науке, технике. Сер. Математика и кибернетика. М.: Знание, 1986. № 10. 48 с.

ПРОГРАММНЫЙ КОМПЛЕКС ФОРМИРОВАНИЯ ОБЛИКА ЛЕТАТЕЛЬНЫХ АППАРАТОВ

Л.Л. Вышинский, О.С. Самойлович, Ю.А. Флёров

Наиболее ответственным этапом проектирования современных летательных аппаратов является предварительное проектирование. На этом этапе при сравнительно малых затратах (5-6% от стоимости опытно-конструкторские работы) принимается до 70% наиболее ответственных решений, определяющих дальнейшую судьбу проекта. Причём эти решения принимаются в условиях неполной информации не только в отношении реализации требований, но и в отношении обоснованности самих требований. (По оценке корпорации РЭНД, неопределённости тактико-технических требований (ТТТ) по системам оружия обуславливают до 30% перерасходов по сравнению с первоначальными сметными оценками, что составило в бюджете США за период 1965-1970 более 10 млрд. долларов.)

В связи с этим разработка автоматизированных систем концептуального проектирования летательных аппаратов (ЛА) представляет собой задачу повышенной важности. Основной целью предварительного проектирования является определение оптимального облика изделия. Для достижения этой цели необходим анализ большого числа альтернативных вариантов ЛА, а также совершенствование методов принятия решений, в том числе определение самого понятия оптимальности для задач проектирования многоцелевых летательных аппаратов. Для автоматизации этих работ и предназначен программный комплекс ОБЛИК, о котором говорится в настоящей статье.

Обычно круг задач, решаемых на начальных этапах проектирования, не является фиксированным, а зависит от специфики проектируемого изделия и от принятой в конструкторском бюро технологии. Тем не менее, можно указать некоторые " типовые " задачи, которые, как правило, возникают при формировании облика. Такими задачами являются:

- анализ тактико-технических требований, оценка их непротиворечивости и реализуемости, наличие прототипов;
- выбор концепции компоновочной схемы;
- определение структуры и состава конструктивных параметров и характеристик изделия;
- построение основных геометрических формообразующих элементов (форма в плане, профили несущих поверхностей, сечения фюзеляжа в районах основных силовых элементов или границ отсеков);
- построение директивной конструктивно-силовой схемы изделия;
- согласование двигателей с планёром и ряд других задач.

Для проверки соответствия конкретной компоновки техническому заданию необходимо уже на этапе формирования облика оценивать аэродинамические, лётно-технические и другие характеристики ЛА.

Под решением перечисленных здесь задач (как, впрочем, и задач, не вошедших в этот перечень) понимается оптимальный выбор конечного числа альтернатив из множества допустимых конструкций, поэтому при постановке конкретной задачи необходимо указать критерий (или критерии) оптимальности такого выбора (11).

Заметим, что для частных задач проектирования и модель объекта, и ограничения на его параметры, и критерии оптимальности являются результатом её точной, математической постановки. Перечисленные выше " типовые " задачи формирования облика могут очень широко варьироваться в своей постановке в зависимости от конкретных обстоятельств их возникновения. Поэтому

комплекс ОБЛИК рассматривается не как средство решения фиксированной задачи (или набора таких задач), а как инструментальная система для постановки, отладки и решения достаточно широкого класса проектных задач.

На рис. 1 приведена укрупнённая структура комплекса ОБЛИК. Описание постановки конкретных задач, которое состоит из описания математически моделей объектов проектирования и из описания принципа оптимального выбора, должно осуществляться на формальном, но доступном для пользователя и для ЭВМ языке. Наиболее подходящими для этой цели языками являются непроцедурные логические языки высокого уровня. В комплексе ОБЛИК используется входной язык системы ФАКИР ([2]) а сама система ФАКИР входит в состав инструментальных средств комплекса.

База знаний МОДЕЛИ представляет собой совокупность описаний взаимосвязанных структур моделей объектов и задач, которые используются при формировании облика. Каждая такая модель представляет структуру данных (конструктивных параметров и характеристик) этой модели, а также связи и ограничения, накладываемые на них. Связи могут носить аналитический, алгоритмический или программный характер. База знаний пополняется за счёт поступающих внешних описаний, а её внутренняя структура строится в результате обработки входных описаний системой ФАКИР и другими инструментальными средствами.

Основными моделями, используемыми в комплексе, являются:

- модели различных компоновочных схем ЛА;
- модели отдельных агрегатов;
- геометрические модели формообразующих элементов планёра;
- модели поставляемых агрегатов;
- упрощённые вычислительные модели расчёта характеристик летательных аппаратов и др.

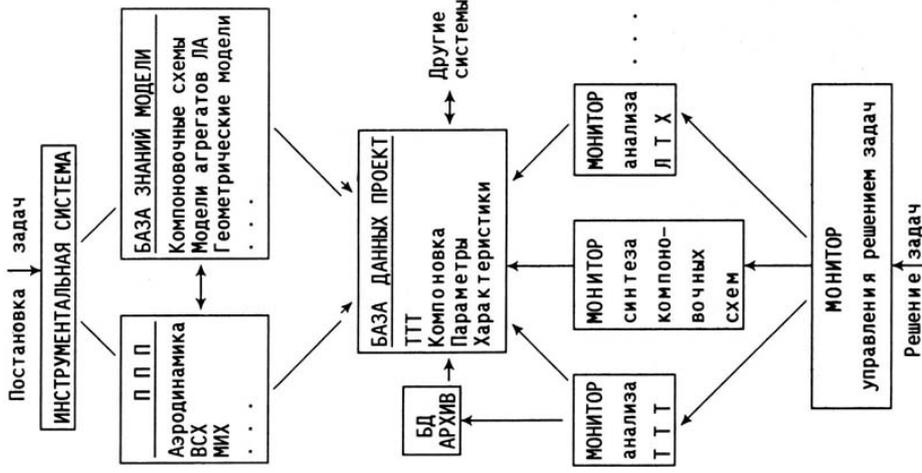


Рис. 1.

Упрощённые расчётные модели служат, как правило, для предварительных или оперативных (быстрых) оценок характеристик. Более тщательный анализ объекта проектирования требует специальных программных средств, которые обычно разрабатываются специалистами соответствующих областей – аэродинамиками, прочнистами, двигателями и т.д. Такие программы объединяются в пакеты прикладных программ (ППП).

В процессе проектирования решается множество задач. Результаты решения в общем случае изменяют текущее состояние проекта и поэтому должны быть зафиксированы. Для этих целей в комплексе ОБЛИК используется база данных ПРОЕКТ, в которой отражается весь ход проектирования, начиная от составления и анализа технического задания до выдачи материалов инженерной записки и технического предложения. База данных ПРОЕКТ хранит также альтернативные решения и хронологию получения тех или *n* результатов.

Для обеспечения решения задач данными о возможных прототипах, характеристиках и конструкции, а также другой фактографической информации служит база данных АРХИВ.

База знаний комплекса ОБЛИК может постоянно расширяться, некоторые модели в процессе работы могут претерпевать существенные изменения или даже вообще уничтожаться. Однако неотъемлемой частью комплекса является некоторая базовая система моделей, которая служит основой при построении различных модификаций и при настройке их для решения конкретных задач. Разработка базовой системы моделей, кроме практического значения, представляет и методический интерес. Поэтому опишем некоторые модели формирования облика из базовой системы.

В практической деятельности КБ наиболее ответственной является задача синтеза компоновочной схемы будущего летательного аппарата, целью

которой является определение значений основных конструктивных параметров проектируемого ЛА. Для решения этой задачи необходимо иметь формализованное описание компоновочной схемы, под которой понимается множество допустимых конструкций в пространстве параметров, определяемое компоновочными связями и всякого рода ограничениями.

Так как размерность пространства конструктивных параметров даже для начальных этапов проектирования достаточно велика, то целесообразно модель компоновочной схемы ЛА разбить, по крайней мере, на два уровня. На верхнем уровне можно представить ЛА небольшим набором обобщённых параметров, а также структурной схемой компоновки ЛА из своих агрегатов. На втором уровне должны быть описаны компоненты-агрегаты ЛА и способы их объединения. Естественно, эти модели связаны между собой как "по вертикали", так и "по горизонтали".

Под описанием модели здесь будет пониматься указание имени моделируемого объекта, перечня внешних параметров, перечня конструктивных параметров и характеристик, состава агрегатов или подмоделей и связей между всеми этими компонентами модели. Если некоторые связи носят программный характер, то в модели должны быть ссылки на соответствующие программные модули.

Здесь будут рассмотрены модели летательного аппарата верхнего уровня, модели основных агрегатов, их компоновка и конструкция. Естественно, что эти модели отвечают требованиям начальных этапов проектирования.

Итак, на верхнем уровне ЛА определяется рядом обобщённых конструктивных параметров:

ЛЕТАТЕЛЬНЫЙ АППАРАТ ::=

Внешними параметрами на начальных этапах проектирования являются ТТТ, среди которых ограничения на лётные характеристики, состав и параметры целевых оборудования и нагрузки, эксплуатационные требования и пр.

Конструктивные параметры:

m_0, m_p – взлётная и расчётная массы,

\bar{m}_T, \bar{m}_H – относительная масса топлива и полезной нагрузки,

S – площадь базового крыла,

$S_{\text{ом}}$ – площадь омываемой поверхности,

P_0 – взлётная нагрузка на крыло,

\bar{P} – стартовая тяговооружённость,

$n_{\text{ж}}$ – количество членов экипажа и др.

Основные характеристики ЛА:

C_{x_0} – лобовое аэродинамическое сопротивление,

C_y – коэффициент подъёмной силы,

K_{max} – максимальное аэродинамическое качество,

$C_y^{\text{доп}}$ – максимально допустимый коэффициент подъёмной силы,

ΔC_y – дополнительная подъёмная сила при взлёте,

$P_{\text{max}}, P_{\text{фор}}$ – максимальная и форсажная стендовые тяги,

C_e – удельный расход топлива на расчётном режиме и др.

Приведённые характеристики являются функциями фазовых переменных скорости, высоты и т.д.

Состав агрегатов летательного аппарата:

A1 – планёр,

A2 – силовая установка,

A3 – самолётные системы,

A4 – целевое оборудование,

A5 – полезная нагрузка,

A6 – снаряжение.

(Модели этих агрегатов будут описаны позднее.)

Основные отношения, связывающие перечисленные величины:

$$m_0 = \sum_{i=1}^6 A_i \cdot m; \quad P_0 = m_0 g / S; \quad \bar{p} = P_{\text{фор}} / m_0 g; \quad m_p = m_0 (1 - 0.5 \bar{m}_T);$$

$$\bar{m}_T = A5 \cdot m / m_0; \quad \bar{m}_T = A2; \quad m_T / m_0; \quad \dots$$

A_i , m – это массы агрегатов ЛА, которые определяются из соответствующих моделей.

Между характеристиками ЛА, параметрами и требованиями ТТТ существуют определённые соотношения, которые здесь будут приведены в общем виде:

$$\bar{f}(TTT, m_0, \bar{m}_T, S, C_{x_0}, K_{\text{max}}, \dots) \geq 0.$$

Эти ограничения образуют так называемые "области существования", которые определяют диапазоны значений основных конструктивных параметров и требования к уровню аэродинамических, высотно-скоростных, дроссельных и других характеристик.

МОДЕЛИ АГРЕГАТОВ ПЛАНЕРА

Планёр характеризуется рядом конструктивных параметров и соотношений такого же вида, что и модели верхнего уровня. А состав его агрегатов включает крыло, фюзеляж, оперение и взлётно-посадочные устройства.

КРЫЛО ::= (см. рис. 2)

Внешние параметры: x , y , z – положение крыла в системе координат ЛА.

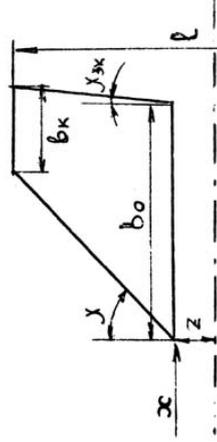


Рис. 2

Конструктивные параметры:

$$S, b_0, b_k, l, \chi, \lambda, \eta, \chi_{\text{эк}}, \bar{C}, T_{\text{ст.проф.}}, \dots$$

Характеристики:

$S_{\text{ом}}$ — площадь омываемой поверхности,

$S_{\text{мид}}$, $X_{\text{мид}}$ — площадь и положение миделя крыла,

$V_{\text{пол}}$ — полезный объём под топливо,

m , $X_{\text{цт}}$ — масса и положение центра тяжести и пр.

Связи и компоновочные ограничения:

$$S = (b_0 + b_k) l / 2;$$

$$\lambda = l^2 / S;$$

$$b_0 - b_k = (\text{tg } \chi - \text{tg } \chi_{\text{эк}})(0.5l - z);$$

$$S_{\text{ом}} = 2S(1 + k\bar{c});$$

Возможно также дальнейшее разбиение крыла на свои агрегаты (наплыв, кессон, закрылки, элероны, отсеки крыла и т.п.).

ФЮЗЕЛЯЖ ::=

Конструктивные параметры:

$$l_{\text{ф}}, d_{\text{пах}}, h_{\text{пах}}, d_{\text{ф}}^{\text{экв}}, \lambda_{\text{ф}}, S_{\text{дон}}, \dots$$

Характеристики:

$S_{\text{ом}}$ — площадь омываемой поверхности,

$S_{\text{мид}}$, $X_{\text{мид}}$ — площадь и положение миделя фюзеляжа,

$V_{\text{пол}}$ — полезный объём под оборудование и нагрузку,

V_T — полезный объём под топливо,

m , $X_{\text{цт}}$ — масса и положение центра тяжести и пр.

Разбиение фюзеляжа на агрегаты носит довольно условный характер. Скорее это не агрегаты, а функционально выделенные части:

ГЧ — головная часть фюзеляжа, предназначенная для размещения оборудования и кабины экипажа. Она разбивается на ряд отсеков: обтекатель,

предкабинный отсек, кабина с подкабинной частью и фонарём, закабинный отсек.

ЦЧ — центральная часть фюзеляжа тоже состоит из ряда отсеков: топливных, ниши шасси, размещения нагрузки, возможно, мотоотсеков и т.п.

ХЧ — хвостовая часть фюзеляжа содержит мотоотсеки, возможно, топливные баки и другие объёмы.

ВЗ — воздухозаборники (если двигатели размещаются в фюзеляже).

Связи и компоновочные ограничения:

$$m = \Gamma Ч. m + ЦЧ. m + ХЧ. m + ВЗ. m;$$

$$m x_{цч} = \sum_{i=1}^4 A_i. m \times A_i. x_{цч}; \quad (A_i \in \{\Gamma Ч, ХЧ, ЦЧ, ВЗ\});$$

$$\lambda_{\phi} = l_{\phi} / d_{\phi}^{экв}; \quad d_{\phi}^{экв} = \sqrt{4 S_{млд} / \pi};$$

$$l_{\phi} = \Gamma Ч. l + ЦЧ. l + ХЧ. l; \dots$$

Многие ограничения определяются на уровне описания моделей агрегатов фюзеляжа, а условия их объединения диктуются выбором принципиальной компоновочной схемы, о которой будет сказано позже.

Аналогичным образом определяются остальные агрегаты планёра (горизонтальное и вертикальное оперение, взлётно-посадочные устройства).

Важную роль при формировании облика играет модель силовой установки.

СИЛОВАЯ УСТАНОВКА ::=

Конструктивные параметры:

$n_{дв}$ — количество двигателей,

$n_{мо}$ — количество мотоотсеков.

Характеристики:

$m, x_{цч}$ — масса и положение центра тяжести,

V_{cy} — полный объём, занимаемый силовой установкой.

Высотно-скоростные характеристики были упомянуты ранее.

Агрегаты силовой установки:

A1 — двигатели,

A2 — воздухоподводящие устройства,

A3 — системы силовой установки,

A4 — топливо.

Связи и компоновочные ограничения:

$$m = n_{дв} A1. m + A2. m + A3. m;$$

$$m x_{цч} = A1. m \times A1. x_{цч} + A2. m \times A2. x_{цч} + A3. m \times A3. x_{цч};$$

$$m_{т} = A4. m; \quad V = A1. V \times n_{дв} + A2. V + A3. V.$$

Общее и целевое оборудование, полезная нагрузка и снаряжение представляются собой (с точки зрения формирования облика ЛА) грузы и объёмы, иногда внешние подвески. Поэтому модель любого из таких агрегатов может быть представлена следующим образом:

РАЗМЕЩАЕМЫЙ АГРЕГАТ ::=

Внешние параметры: x, y, z — положение агрегата в системе координат ЛА.

Конструктивные параметры:

$m, x_{цч}, V, H, B, L (H, B, L - \text{габариты}).$

Привязка размещаемых агрегатов к конкретным объёмам или отсекам планёра должна быть описана либо непосредственно при упоминании модели летательного аппарата, либо в специальном отношении, которое определяет правила внутренней компоновки ЛА.

Аэродинамические и конструктивные свойства ЛА в значительной степени зависят от принципиальной компоновочной схемы, которая задаёт размещение основных агрегатов относительно друг друга. Для описания некоторого компоновочных схем необходимо ввести вектор компоновочных признаков (3), каждый из которых характеризует определённое свойство компоновки

и может принимать конечное число значений. Ниже приведён один из возможных вариантов векторов:

ВЕКТОР КОМПОНОВЧНЫХ ПРИЗНАКОВ ::=

БС — балансировочная схема ЛА (может принимать значения: нормальная, бесхвостка, утка);

ПК — положение крыла по высоте (низкоплан, среднеплан, высокоплан);

P_{10} — размещение горизонтального оперения (хвостовая часть фюзеляжа, носовая часть фюзеляжа, воздуховозборники);

$N_{во}$ — количество килей (один или два);

$P_{во}$ — размещение килей (фюзеляж, крыло, горизонтальное оперение (ГО));

$P_{дв}$ — размещение двигателей (в центральной или в хвостовой части фюзеляжа);

$N_{вз}$ — количество воздуховозборников;

$P_{вз}$ — расположение воздуховозборников (лобовое, боковое, подфюзеляжное, верхнее);

КС — компоновка сопел (сомкнутые, разнесённые);

$P_{кш}$ — размещение колёс главных стоек шасси (в фюзеляже, в крыле);

НУШ — направление уборки главных стоек шасси (вперёд, назад, в стороны, под себя);

$P_{эк}$ — размещение экипажа (если $n_{эк} > 1$; тандем, рядом).

Множество допустимых компоновочных схем определяется рядом условий совместимости тех или иных значений приведённых признаков. Эти условия записываются в виде некоторого логического выражения. Например:

Если БС = нормальная, то P_{10} = хвостовая часть фюзеляжа; если БС = утка, то P_{10} = носовая часть фюзеляжа или воздуховозборник; если БС = бесхвостка, то P_{10} теряет смысл, т.е. может быть произвольным; если $N_{во} = 1$, то $P_{во}$ = фюзеляж и т. д.

Компоновочные признаки могут использоваться в моделях агрегатов для определения условий их стыковки и взаимного расположения.

Приведённые здесь модели объекта проектирования необходимо дополнить указанием способа расчёта основных характеристик ЛА. Как правило, эти характеристики вычисляются с помощью специальных программных модулей, которые представлены в соответствующих ППП. Выбор тех или иных средств для расчёта характеристик зависит от типа ЛА, характера модели, области применимости, необходимых точностей и имеющихся временных и других ресурсов.

Базовые модели объектов проектирования имеют, как правило, много степеней свободы. При постановке конкретной задачи модель уточняется, т.е. вводятся дополнительные условия, связи и отношения между параметрами, некоторые из них принимают фиксированные значения, другие в той или иной форме объявляются варьируемыми. Количество варьируемых параметров определяет размерность задачи, которая лимитируется лишь соображениями реальности её решения. Постановка завершается указанием критериев оптимизации. Иногда размерность не позволяет организовать оптимизацию за доступное время, что приводит к необходимости декомпозиции и моделей объектов, и самих проектных задач. Постановка задачи в ЭВМ осуществляется теми же языковыми средствами, что и описание моделей. Ниже дан пример типичной постановки задачи формирования облика.

ЗАДАЧА ФО ЛА ::=

Внешние параметры (ТТТ):

$$L_0, L_n, V_{\text{пмах}}, L_{\text{разб}}, \Psi_{\text{разв}}, t_{\text{разв}}, \dots$$

Модель объекта:

ЛА — летательный аппарат, где u вектора компоновочных признаков:

БС = нормальная, ПК = среднеплан, P_{10} = хвостовая часть фюзеляжа, $N_{во} = 1$,

$N_{дв} = 1, P_{дв} =$ в хвостовой части фюзеляжа, $N_{вз} = 1, P_{вз} =$ подфюзеляжное, $P_{кш} =$ в крыле, НУШ = в стороны.

Кроме того, выполняется ряд дополнительных условий:

двигатель фиксирован, оборудование размещается в отсеках носовой части фюзеляжа, топливо — в центральной части фюзеляжа, а также:

$$n_{ж} = 1, A5. m = m_5, A4. m = m_4, \dots$$

Варьируемые параметры:

$$S \in [S_{мин}, S_{макс}]; \lambda \in [\lambda_{мин}, \lambda_{макс}]; \chi \in [\chi_{мин}, \chi_{макс}]; \dots$$

Критерии оптимизации:

$$V_y \rightarrow \max; R_{разг} \rightarrow \min; m_0 \rightarrow \min.$$

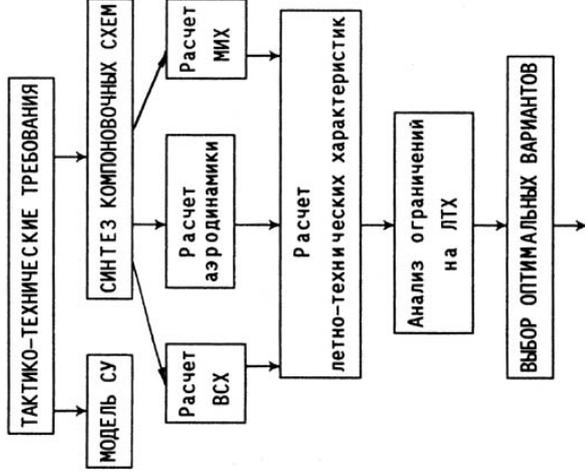


Рис. 3

Правильно составленное описание модели объекта и задачи проектирования средствами инструментальной системы автоматически преобразуются в программу решения поставленной задачи. В общем виде блок-схема этого решения представлена на рис. 3. Заметим, что такая схема принципиально подходит для любой из перечисленных в начале содержательных задач. Основная разница состоит в блоке синтеза компоновочных схем. Именно в этот блок преобразуется настроенная модель объекта. Здесь происходит вычисление конструктивных параметров, проверка всех ограничений и отбраковка вариантов, не прошедших эти ограничения. Для прошедших вариантов осуществляется расчёт характеристик и проверка ограничений, заданных в ТТТ. Из оставшихся компоновок отбираются наилучшие в смысле заявленных критериев оптимальности.

По такой схеме решаются практически все задачи формирования облика. Однако уже при 5-8 варьируемых параметрах количество анализируемых вариантов может равняться тысячам. Поэтому решение таких задач обычно длится несколько часов и должно вестись в пакетном режиме. Но прежде чем запустить трудоёмкую задачу, необходимо удостовериться в её качественной постановке: адекватности модели, разумно выбранных диапазонов варьируемых параметров, проанализировать некоторые частные решения и т.п. Для этих целей, а также для анализа и интерпретации результатов необходим диалоговый режим решения задач.

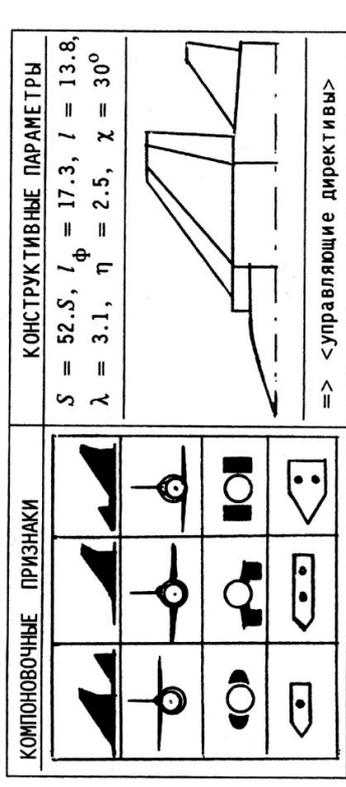
Работа с объектами в диалоговом режиме предусматривает возможность задавать и изменять структурную схему ЛА, вектор признаков компоновочной схемы, значения параметров, проводить расчёты и анализировать их, осуществлять параметрические исследования, которые не требуют больших временных затрат. Такие диалоговые процедуры должны создаваться в процессе описания моделей и задач, поскольку заранее нельзя предусмотреть все

возможные сценарии работы конструктора. Особое значение при этом имеет форма представления информации, и поэтому для разработки управляющих диалоговых мониторов в комплексе ОБЛИК предусмотрены специальные инструментальные средства [4].

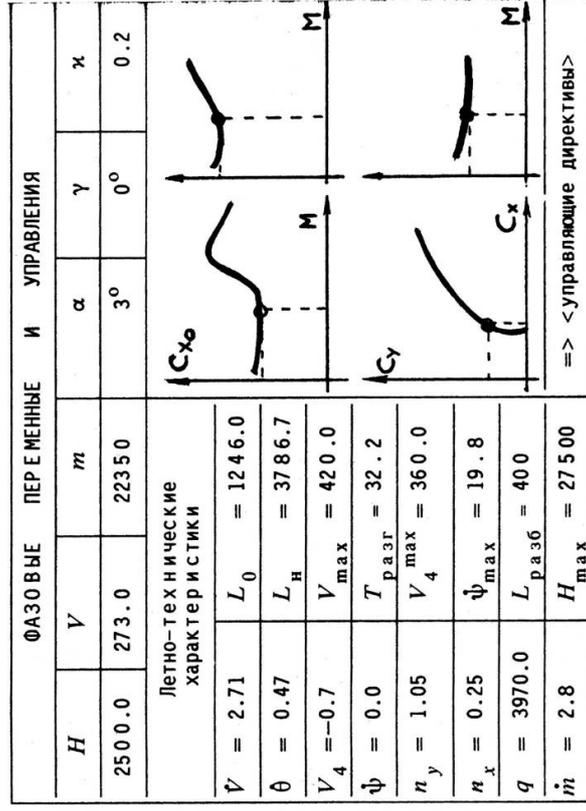
Диалогом обеспечивается также процесс анализа решений задач проектирования, полученных в пакетном режиме, в частности принятие решения о выборе перспективных вариантов для дальнейшей проработки из всего множества решения.

В комплексе ОБЛИК, наряду с базовыми моделями, реализуются некоторые сценарии работы с ними. Они реализованы в специальных диалоговых программах: мониторе построения области существования ЛА, мониторе синтеза компоновочной схемы, мониторе анализа компоновочной схемы и пр. На рис. 4 приведена форма представления информации в этих программах. Приведённые здесь числовые и графические поля отражают процесс решения задачи, а управление решением осуществляется либо набором директив, либо указанием необходимого пункта предъявляемого меню.

Как уже говорилось, результат решения задач проектирования помещается в БД **ПРОЕКТ**. Логическая схема БД должна позволять реализовывать структуры данных, соответствующие приведённому выше математическим моделям. Однако эти модели являются базовыми, т.е. могут изменяться в процессе их использования. Поэтому логическая схема БД **ПРОЕКТ** не должна зависеть от переименования параметров, изменения их количества и организации иерархии.



а



б

Рис. 4: а – монитор синтеза компоновочных схем;
б – монитор анализа лётно-технических характеристик.

На рис. 5 показана структура иерархически организованных объектов в базе данных. Кроме самих объектов, здесь должна храниться информация о принадлежности объекта к той или иной теме, о задачах, при решении которых получены данные, об источниках входной информации для этих задач, перечень альтернативных вариантов решения, а также сведения принадлежности конкретным объектам и данным к варианту или группе вариантов. Такие структуры позволяют не только хранить результаты решения всех задач, но и иметь хронологию развития проекта, а также отслеживать различные изменения, итеративные циклы в проектировании, возврат к ранним вариантам тех или других агрегатов и т.п. При этом основными задачами управления проектированием являются: поддержание логической целостности данных, своевременная модификация исходных позиций при решении проектных задач и выдача сообщений об изменениях в проекте.

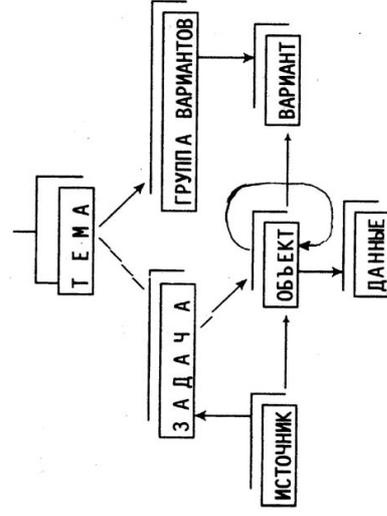


Рис. 5

В этой статье была затронута лишь небольшая часть вопросов, возникших при разработке комплекса ОБЛИК, но хотелось бы подчеркнуть, что при использовании САПР для решения сложных концептуальных проектных проблем

основная тяжесть ложится на инструментальные и языковые средства описания построения формальных математических моделей.

Литература

1. Краснощёков П.С., Морозов В.В., Фёдоров В.В. Последовательное агрегирование в задачах внутреннего проектирования технических систем // Изв. АН СССР. Сер. Тех. киберн. 1979. № 5. С. 5-12.
2. Вышинский Л.Л., Прибытков Ю.Д., Шиленко В.И., Широков Н.И. Инструментальная система ФАКИР // Изв. АН СССР. Сер. Техн. киберн. 1986, № 3. С. 35-40.
3. Езер С.М., Лисейцев Н.К., Самойлович О.С. Основы автоматизированного проектирования самолётов. М.: Машиностроение, 1986.
4. Вышинский Л.Л., Гринёв И.Л., Шиленко В.И., Широков Н.И. Инструментальные средства САПР. Наст. сб.

СТРУКТУРА МОДЕЛЕЙ В ЗАДАЧАХ ПРОЕКТИРОВАНИЯ

Л. Л. Вышинский

Одна из функций систем автоматизированного проектирования состоит в технологической поддержке создания математических моделей объектов проектирования. Однако программная реализация таких функций требует изучения основных структур, которые используются при построении моделей объектов и задач определённых предметных областей. Именно этим аспектам применительно к изданию машиностроения посвящена настоящая работа.

Будем рассматривать математические модели объектов проектирования (ММОП) в виде множества векторов конструктивных параметров изделия, заданного следующим образом:

$$X(y) = \{x = \langle x_1, \dots, x_n \rangle \mid x_i \in X_i, i = 1, \dots, n; P(x, y)\}, \quad (1)$$

где X — наименование модели объекта (множества), y — вектор внешних параметров, от которых зависит X , x — вектор конструктивных параметров, $x_i, i = 1, \dots, n$, — конструктивные параметры, X_i — множества значений параметра x_i , $P(x, y)$ — предикат, задающий связь между параметрами.

Множества X_i являются либо примитивными множествами (т.е. действительными или целыми числами (E, I), логическими значениями (L) и др.), либо представляют собой модели агрегатов объектов, т.е. имеют вид, аналогичный (1):

$$X_j(y) = \{x_j = \langle x_{j_1}, \dots, x_{j_n} \rangle \mid x_{j_i} \in X_{j_i}, j = 1, \dots, n; P_j(x_j, y)\},$$

Под решением задачи в проектировании обычно понимается целенаправленный выбор одного или нескольких экземпляров векторов конструктивных параметров из множества допустимых конструкций ММОП.

Решение, как правило, определяется рядом ограничений на характеристики объекта и некоторым понятием оптимальности, например, скалярным или векторным критерием, отношением предпочтения и т.п. В любом случае множество выбора определяется таким же образом, что и в (1):

$$W(y) = \{w \mid w \in X(y); P_w(w, y)\}, \quad (2)$$

где w — вектор решения задачи, P_w — предикат, определяющий условие выбора.

Существенным отличием W от ММОП является требование конечности множества выбора. Это требование должно быть учтено при построении P_w . В частности, если $X(y)$ имеет бесконечное число элементов, а задача состоит в нахождении паретовского множества для некоторого векторного критерия, то множество выбора может оказаться также бесконечным. В таких случаях приходится рассматривать не всё множество, а некоторое его подмножество, например, его конечно-точечную аппроксимацию.

Совокупность описаний X, X_1, \dots, X_n определяет структуру ММОП. Специально оговорим недопустимость прямой рекурсии в этой структуре, а для реализации моделей, в которых рекурсия является, по существу, необходимой, воспользуемся известным механизмом ссылок. Будем считать, что при описании любого множества X одновременно определяется новое примитивное множество $Ref(X)$. Каждый элемент этого множества представляет ссылку (адрес, номер, идентификатор и т.п.) на некоторый элемент множества X . Кроме того, каждому ссылочному множеству принадлежит так называемая пустая ссылка — λ , которая не соответствует ни какому элементу X . Ссылочные множества и ссылочные переменные могут использоваться при описании конструктивных параметров рекурсивных ММОП.

Введём понятие типа τ множества X :

$$\tau(X) = \{\tau(X_1), \dots, \tau(X_n)\},$$

при этом для примитивных множеств тип определяется следующим образом:

$$\tau(E) = E, \tau(I) = I, \tau(L) = L, \tau(Ref(z)) = Ref(z), \dots$$

Поскольку прямая рекурсия запрещена, то $\tau(X)$ соответствует конечному фиксированному дереву, терминальные вершины которого помечены наименее примитивными множествами.

Рассматривая структуру предикатов в ММОП, в первую очередь необходимо описать простейшие, атомарные предикаты. Атомарными предикатами в ММОП являются:

- параметры и константы логического типа,
- арифметические или другие примитивные отношения ($=, <, >, \dots$),
- программные отношения ($M(w_1, \dots, w_k)$), где M — имя отношения $w_i \in W_i$, $i = 1, \dots, k$, а $\tau(W_k)$ — характеристика M .

Любой предикат P в ММОП образуется из атомарных с помощью логических операций \vee, \wedge, \neg и кванторов существования (\exists) и всеобщности (\forall).

Поскольку $\tau(X)$ представляет собой конечное дерево, то оно может быть "развёрнуто" по терминальным вершинам: $\tau(X) \sim (\tau_1, \dots, \tau_N)$, где $\tau_i \in \{E, I, L, \dots\}$, (в том числе τ_i могут быть ссылочными типами). В соответствии с этим может быть развёрнут и вектор x :

$$x \sim \langle r_1, \dots, r_N \rangle, r_i \in R_i, \tau(R_i) = r_i.$$

Тогда в пространстве примитивных конструктивных параметров R^N множество выбора определяется следующим характеристическим предикатом:

$$\chi_W(x | y) = P_W(x, y) \wedge P(x, y) \wedge \chi_1(x_1 | y) \wedge \dots \wedge \chi_n(x_n | y),$$

где χ_i — характеристические предикаты моделей X_i . Если X_i — примитивное множество, то $\chi_i = \text{true}$.

Таким образом, дерево $\tau(X)$ и формула χ_W представляют собой структуру математической модели задачи проектирования.

Для конструктивного получения элементов множества W необходимо по предикату χ_W построить алгоритм решения поставленной в (2) задачи. Если

разработка ММОП и описание множества выбора в значительной мере были неформальными процессами, то алгоритмизация решения задачи во многих практических случаях может быть полностью автоматизирована. Практически необходимо построить вычислительную модель решения уравнения $\chi_W = \text{true}$ относительно x . В зависимости от конкретного вида предиката χ_W эта модель может быть либо аналитической формулой, либо последовательностью операторов, дающей полное решение, либо, в худшем случае, численной итерационной процедурой частичного или приближённого решения поставленной задачи. Каждая вычислительная модель характеризуется своими входами и выходами, или, другими словами, вычислительной схемой.

Вычислительной схемой для произвольного предиката $P(R)$, где $R = r_1, \dots, r_n$ — совокупность его переменных, будем называть разбиение R на два непересекающихся подмножества: подмножество входных переменных R_{in} и подмножество выходных переменных R_{out} таких, что известен алгоритм, дающий полное множество решений уравнения

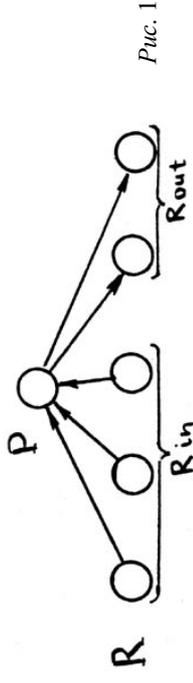
$$P(R) = P(R_{in} | R_{out}) = \text{true}$$

относительно переменных из R_{out} при любых значениях переменных из R_{in} . Вычислительную схему будем обозначать в виде пары

$$s = (R_{in}, R_{out}), \text{ где } R_{in} \cup R_{out} = R, R_{in} \cap R_{out} = \emptyset.$$

Множество решений $\{\bar{r}_{out}\}$ должно быть конечным, но может быть и пустым.

На рис. 1 изображена вычислительная схема предиката, принадлежность переменных к входным или выходным множествам соответствует направлению дуг этого графа.



Через $S(P)$ обозначим всё множество вычислительных схем предиката. Заметим, что для любого $P(R)$ разбиение $s^0 = (R, \emptyset)$ всегда является вычислительной схемой и, значит, $s^0 \in S(P)$. Справедливо также следующее утверждение:

$$(R_{in}, R_{out}) \in S(P) \wedge (R_1 \subseteq R_{out}) \rightarrow (R_{in} \cup R_1, R_{out} \setminus R_1) \in S(P).$$

Для построения множества вычислительных схем произвольного предиката необходимо в первую очередь найти все вычислительные схемы его атомарных компонент. $S(P)$ (P – атомарные) зависит от конкретной реализующей среды. В частности, можно считать, что для любых арифметических отношений (кроме равенства) множества \mathcal{S} вырождены до единственной схемы $s^0 = (R, \emptyset)$. На рис. 2а изображена структура этой схемы:

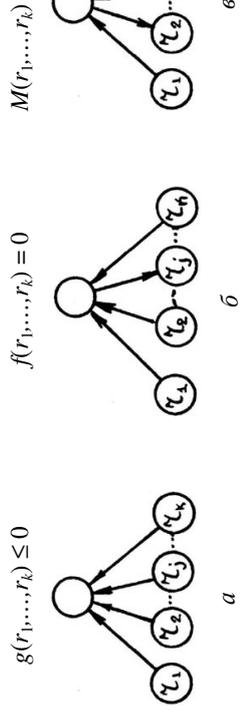


Рис. 2

На рис. 2б дана вычислительная схема арифметического равенства $f(r_1, \dots, r_k) = 0$, которое может быть аналитически разрешимо относительно переменной r_j . Полное множество вычислительных схем для таких равенств представляет собой следующую совокупность:

$$\{(R, \emptyset), \bigcup_j (R \setminus r_j, r_j)\},$$

где объединение берётся по тем j ,

относительно которых это равенство разрешимо.

Что касается предикатов, задаваемых программными отношениями (рис. 2в) множества вычислительных схем определяются спецификациями формальных параметров.

Знание вычислительных схем атомарных подформул позволяет последовательно строить вычислительные схемы для произвольных предикатов.

Пусть $P(R) = \bigvee_{i=1}^k P_i(R^i)$, где $R = \bigcup_{i=1}^k R^i$, и известны $S(P_i)$, $i = 1, \dots, k$.

Тогда можно показать, что

$$S(P) = \bigcup \left(R \setminus \bigcap_{i=1}^k R_{out}^i, \bigcap_{i=1}^k R_{out}^i \right),$$

где внешнее объединение берётся по всем комбинациям

$$(R_{in}^i, R_{out}^i) \in S(P_i), i = 1, \dots, k.$$

В принципе, можно брать лишь в определённом смысле согласованные вычислительные схемы, а именно такие схемы, для которых $R_{out}^i = R_{out}^j$ при всех i и j .

На рис. 3 представлены две структуры вычислительных схем: двудольный граф с множеством вершин $\{P_i\}_i \cup R$, определяющий систему согласованных схем $\{S(P_i)\}_i$, и схема для дизъюнкции. Аналогичные структуры, правила согласования и формулы для построения множества вычислительных схем можно привести для конъюнкции и кванторов существования и всеобщности.

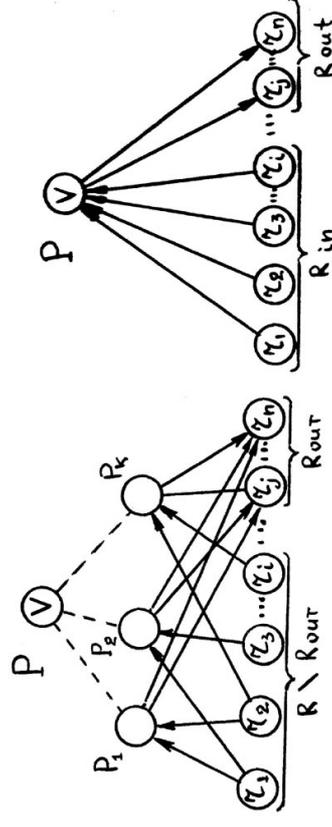


Рис. 3

Пусть $P(R) = \bigwedge_{i=1}^k P_i(R^i)$, где $R = \bigcup_{i=1}^k R^i$, и известны $S(P_i)$, $i = 1, \dots, k$.

Можно показать, что

$$\bigcup \left(R \setminus \bigcap_{i=1}^k R_{out}^i, \bigcap_{i=1}^k R_{out}^i \right) \subseteq S(P),$$

где внешнее объединение берётся лишь по согласованным, но уже в другом смысле комбинациям вычислительных схем $(R_{in}^i, R_{out}^i) \in S(P_i)$. На рис. 4 приведены соответствующие структуры.

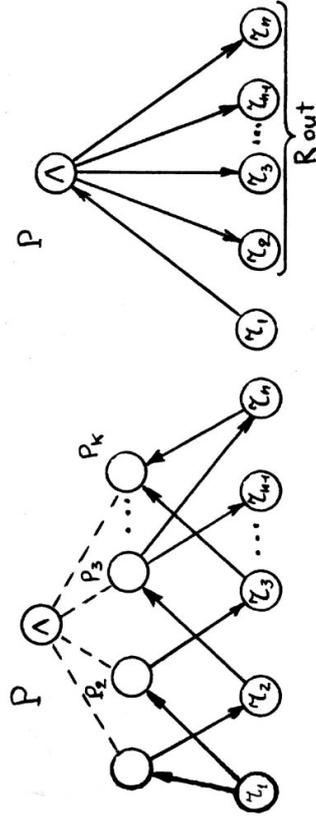


Рис. 4

Под требованием согласованности здесь понимается необходимость выполнения следующих условий:

двудольный граф $\{P_i\}_i \cup R$ не содержит циклов,

$$R_{out}^i \cap R_{out}^j = \emptyset \text{ для всех } i \neq j.$$

Каковы бы ни были P_i , всегда существует для конъюнкции хотя бы одна согласованная совокупность вычислительных схем $\{S(P_i)\}_i$.

Для кванторов существования и всеобщности вопрос о конструктивных алгоритмах вычислений может решаться при одном условии, а именно: множество значений, связанных кванторами переменных, конечно, и существует

эффективный алгоритм его получения.

Если $P(R) = \exists (v \in V) P_1(v, R)$ и множество V конечно, то

$$S(P) = \bigcup (R \setminus R_{out}, R_{out})$$

где объединение берётся по всем $R_{out} \subseteq R$, таким, что

$$(V \cup (R \setminus R_{out}), R_{out}) \in S(P_1).$$

Аналогично, если $P(R) = \forall (v \in V) P_1(v, R)$ и V конечно, то любая вычислительная схема предиката P_1 , имеющая вид $(V \cup R_{in}, R_{out})$, где $R_{in} \cup R_{out} = R$, удовлетворяет условию

$$(R_{in}, R_{out}) \in S(P).$$

Здесь обратное утверждение не проходит, о чём свидетельствует следующий пример:

$$P(x, y) = \forall (v \in \{1, 0\})(x + vy = 1),$$

$$P(x, y) = (x + y = 1) \wedge (x = 1)$$

Следовательно, $(\emptyset, \{x, y\}) \in S(P)$, но

$$(V, \{x, y\}) \notin S(P), \text{ где } P_1 = (x + vy = 1).$$

Для кванторов могут быть построены такие же двухуровневые структуры вычислительных схем, как и для дизъюнкции и конъюнкции.

Таким образом, для произвольного предиката, в соответствии с логической структурой, можно выстроить иерархию двудольных графов, образованных вычислительными схемами соответствующих подформул. При этом, как уже говорилось, на каждом уровне схемы должны быть в определённом смысле согласованными. На рис. 5 сделана попытка графически изобразить такую иерархию. Сплошные дуги на этом графе соответствуют вычислительным схемам, а пунктирные — логической структуре предиката.

Следует обратить внимание на то обстоятельство, что некоторые компоненты рассматриваемых моделей связаны между собой только посредством ссылочных переменных, и поэтому вычислительные модели этих компонент строятся независимо. Однако так или иначе все компоненты модели влияют

друг на друга, и, следовательно, согласование должно проводиться не только в рамках одной компоненты модели, но должно учитывать структуру вычислительных схем других компонент, связанных с ней посредством ссылок.

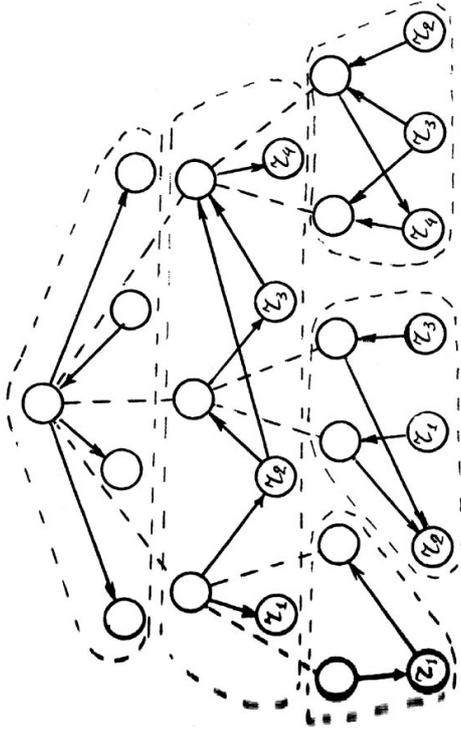


Рис. 5

Для алгоритмизации задачи необходимо построение согласованных на всех уровнях вычислительных схем. Причём желательно найти такое согласование, которое бы привело к минимуму $\|R_m\|$ на верхнем уровне. Если $\|R_m\| = 0$, то это, как правило, означает возможность получения полного решения задачи без применения численных процедур. Описание алгоритмов получения вычислительных моделей выходит за рамки настоящей статьи.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА САПР

Л.Л. Вышинский, И.Л. Гринёв, В. И. Шилленко, Н.И. Широков

Инструментальные программные средства (ИС) в системах автоматизированного проектирования необходимы для автоматизации создания расчётных, информационнох и управляющих программ как при разработке САПР, так и в процессе их использования. Применение ИС позволяет существенно сократить сроки создания прикладных подсистем САПР, впоследствии облегчить различные их модификации. Однако ИС необходимы и в процессе автоматизированного проектирования при постановке конкретных задач. Постановка задач проектирования включает построение моделей объектов проектирования, описание критериев и областей поиска при выборе оптимальных вариантов проекта, описание форм представления текстовой и графической информации о результатах решения, а также установление связей с другими задачами.

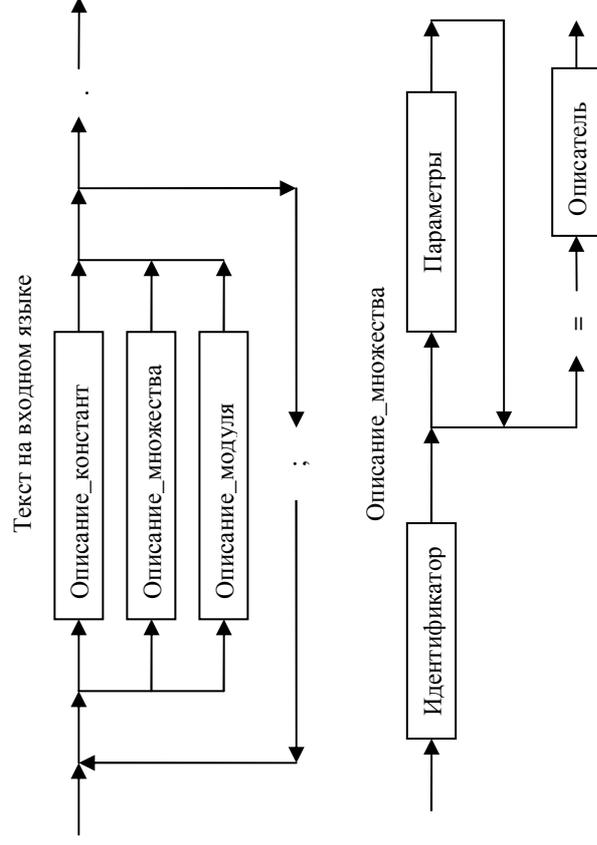
В настоящей статье описывается инструментальная система, реализованная в целях технологической поддержки создания и эксплуатации САПР летательных аппаратов. Одним из основных элементов этой системы является логический, процедурный язык высокого уровня, с помощью которого можно описывать структуру и состав параметров модели, а также ограничения и связи между ними. Формальные описания на этом языке используются для построения моделей в базе знаний некоторой предметной области [1] или, если речь идёт о постановке конкретной задачи, для генерации программы её решения. В качестве генератора расчётных программ в ИС используется система ФАКИР (система формирования алгоритмов конструкторских и инженерных расчётов), которая анализирует исходное описание, строит внутреннее

представление модели, планирует необходимые вычислительные процедуры и создаёт программы, готовые для решения сформулированных задач [2].

Для обеспечения необходимого сервиса при подготовке входной информации и анализе результатов решения в рассматриваемой системе предусмотрены средства организации ввода-вывода данных. Кроме того, в состав инструментальных средств входит система управления базами данных. Задачи САПР предъявляют к СУБД свои специфические требования. Некоторые из них описаны в [3].

Язык описания объектов и задач проектирования

Концептуальной основой используемого здесь языка является представление объекта в виде множества наборов значений его конструктивных параметров или характеристик [1]. Ниже даётся описание синтаксиса и семантики этого языка.



Описание модели на входном языке — это последовательность описаний множеств, снабжённых уникальными именами. В языке считаются пре-дописанными примитивные множества: ВЕЩ — вещественные числа, ЛОГ —

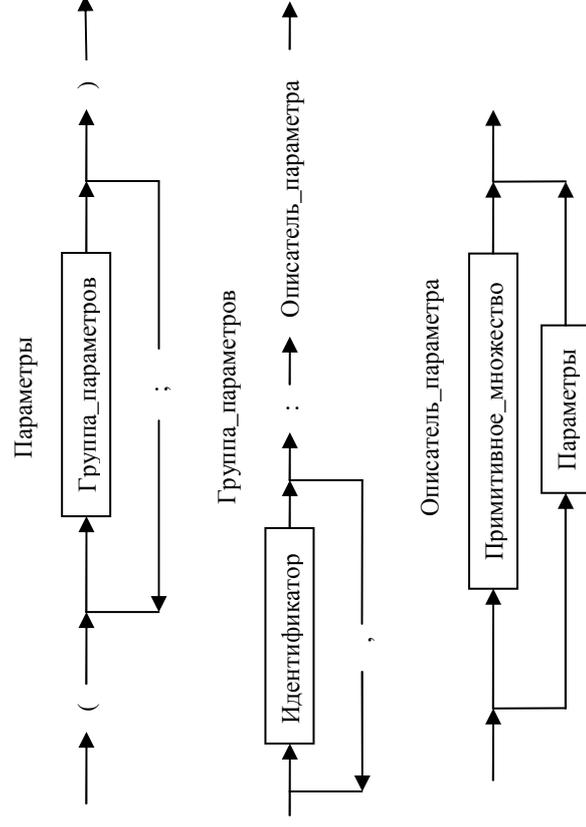
логические значения (ЛОЖЬ, ИСТИНА). Все остальные множества строятся декартовы произведения ранее описанных множеств. Компоненты дека произведения снабжаются именами. Таким образом, описываемое имеет древовидную структуру: терминальные вершины соответствуют примитивным множествам; нетерминальные вершины — промежуточным множествам последовательности описаний; дуги помечаются именами компонент соответствующего декартова произведения.

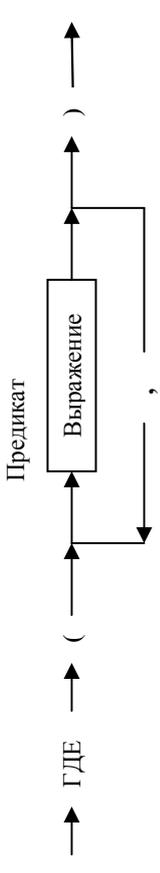
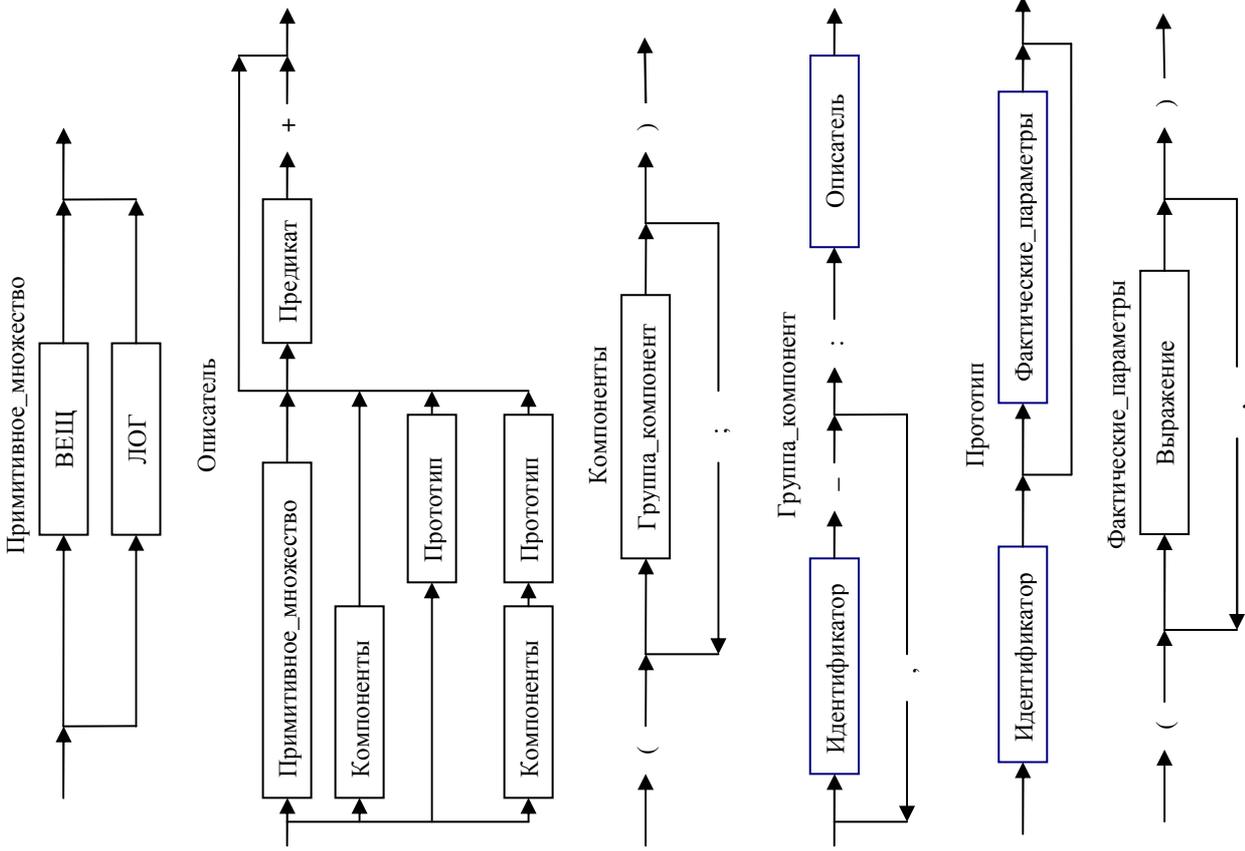
Пример 1.

ТОЧКА = (X, Y: ВЕЩ);

ОТРЕЗОК = (A, B: ТОЧКА).

Здесь описаны два множества — ТОЧКА и ОТРЕЗОК. Множество ТОЧКА — это декартово произведение двух компонент X и Y — примитивных множеств ВЕЩ; множество ОТРЕЗОК имеет две компоненты с именами A и B — ранее описанные множества ТОЧКА.





Использование предикатов в описании множеств даёт возможность рассматривать подмножества декартовых произведений.

Пример 2.

КВАДРАНТ1 = (X, Y : ВЕЩ) ГДЕ (X > 0; Y > 0);
 ОТРЕЗОК1 = (A, B : КВАДРАНТ1)
 ГДЕ (A.X = B.X; A.Y = B.Y - 1).

Здесь множество КВАДРАНТ1 — пары положительных вещественных чисел (X,Y), множество ОТРЕЗОК1 — пары точек из первого квадранта (множество КВАДРАНТ1), являющиеся концами вертикального отрезка единичной длины.

Ниже приводятся синтаксические диаграммы, задающие правила написания предикатов. Предикаты — это логические выражения, составленные из переменных, соответствующих терминальным вершинам в структуре описания множества. Эти переменные записываются с помощью селекторных выражений — последовательностей имён компонент, задающих путь в дереве от корня до соответствующей вершины. В качестве корня дерева берётся то множество, в предикате которого записано рассматриваемое селекторное выражение.

Пример 3.

ОТРЕЗОК1 = (A, B : (X, Y : ВЕЩ) ГДЕ (X > 0; Y > 0))
 ГДЕ (A.X = B.X; A.Y = B.Y - 1).

Здесь описание ОТРЕЗОК1 эквивалентно описанию из примера 2.

С помощью параметров можно описывать семейства множеств.

Пример 4.

ОТР (L : ВЕЩ) = (A, B : (X, Y : ВЕЩ))
 ГДЕ (L.**2 = (A.X - B.X)**2 + (A.Y - B.Y)**2).

Здесь определяется семейство множеств отрезков длины L.

Использование прототипов при описании множеств позволяет экономно задавать различные модификации моделей.

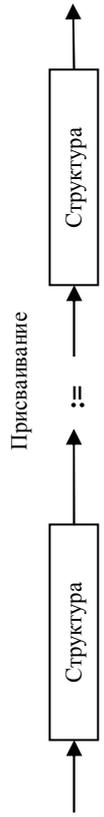
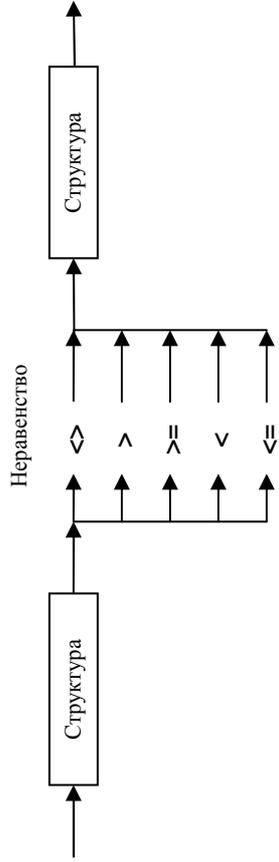
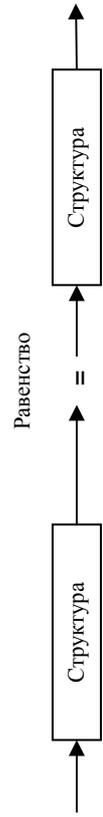
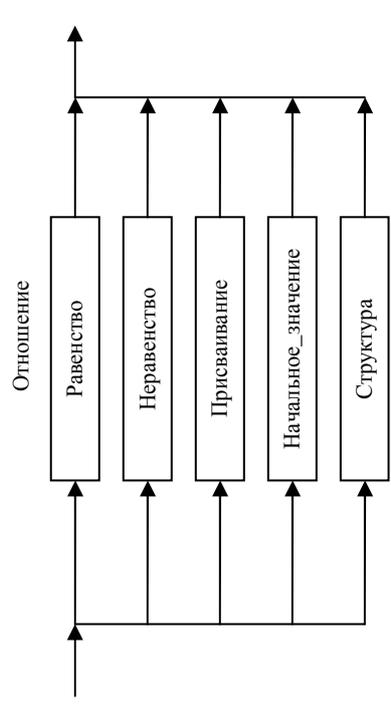
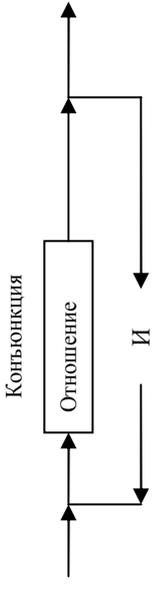
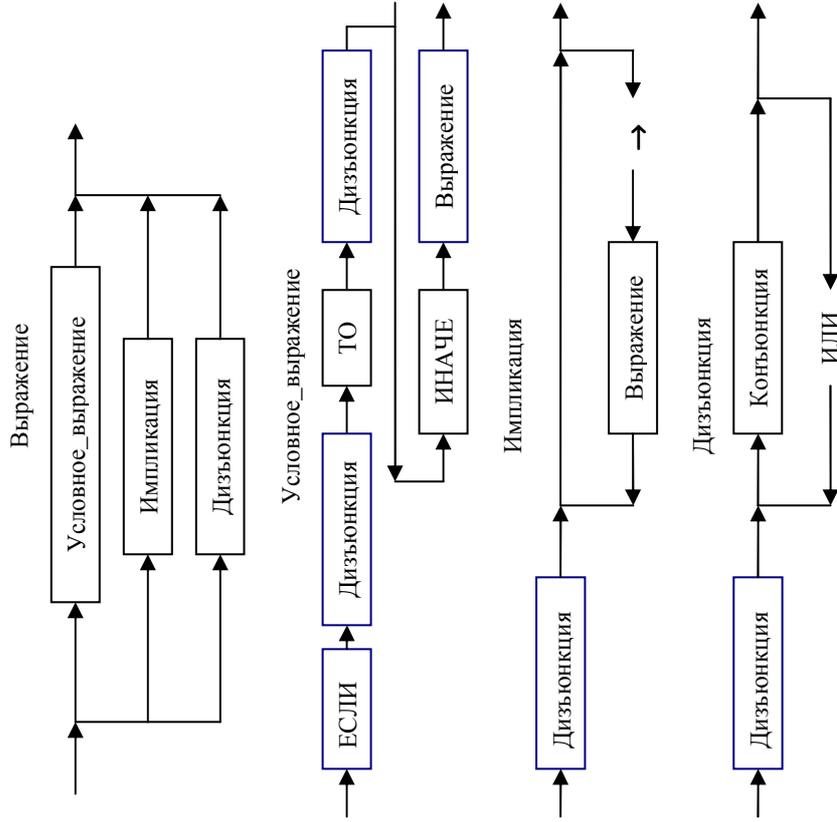
Пример 5.

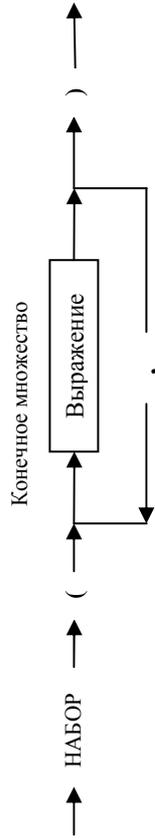
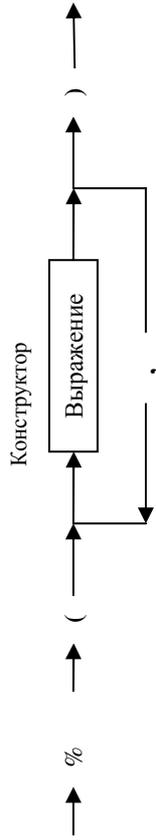
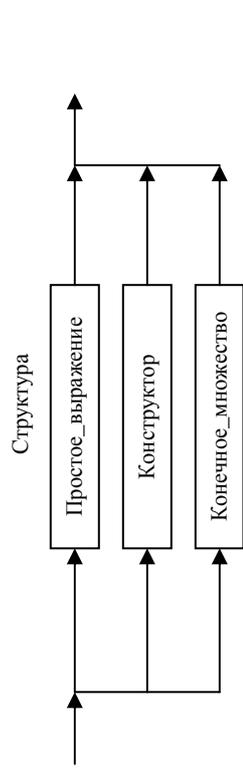
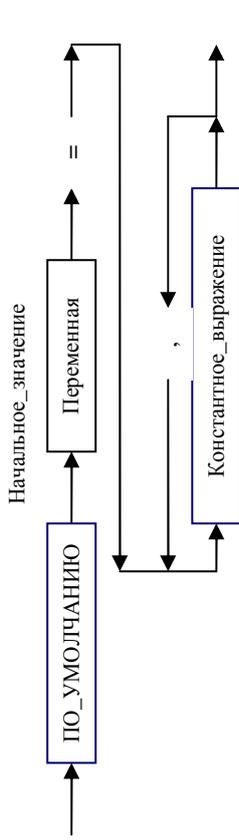
ОТР1 = ОТР (1).

ОТР1 определено как множество отрезков, длина которых равна единице, множество ОТР является прототипом для ОТР1, а выражение в скобках — фактический параметр использованного прототипа. В одном описании допускается одновременное задание компонент как явным образом, так и через прототип.

Пример 6.

ОТР2 = (К : ВЕЩ) ОТР1 (К).





Пример 7.

СЕТКА = (X, Y: ВЕЩ)

ГДЕ (X = НАБОР (10, 11, 12, 13);

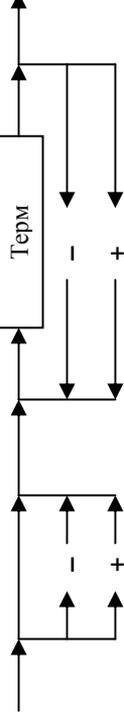
Y = НАБОР (20, 21, 22, 23));

ЧЕТЫРЕ_ТОЧКИ = (X, Y : ВЕЩ)

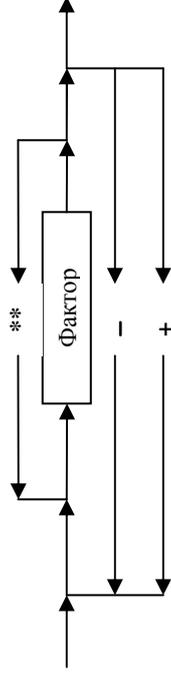
ГДЕ (%(X, Y) = НАБОР (%(10, 20),

% 11, 21), %(12, 22), %(13, 23)).

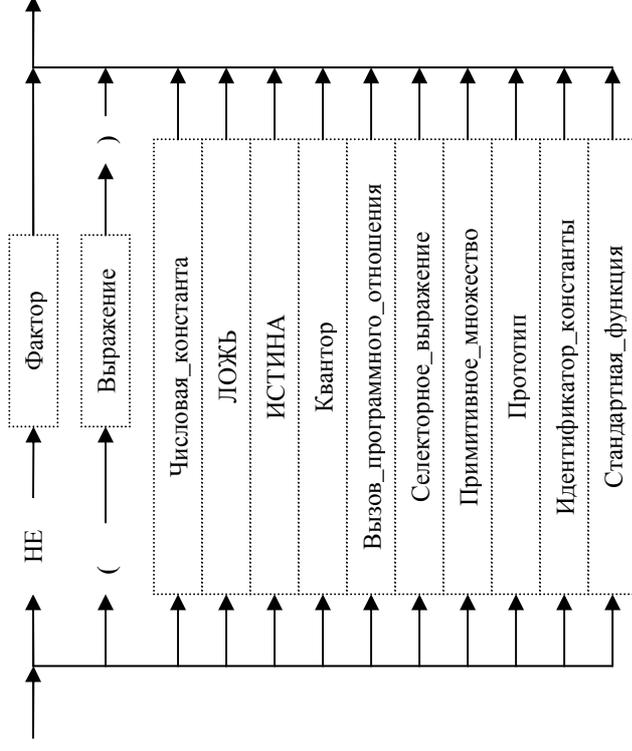
Простое_выражение

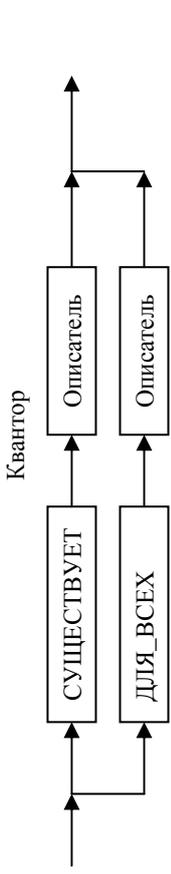


Терм



Фактор





В качестве логических операций допускается использование кванторов существования и всеобщности.

ТОЧКА_НА (M : ОТРЕЗОК) = ТОЧКА

ГДЕ (СУЩ (K : ВЕЩ)

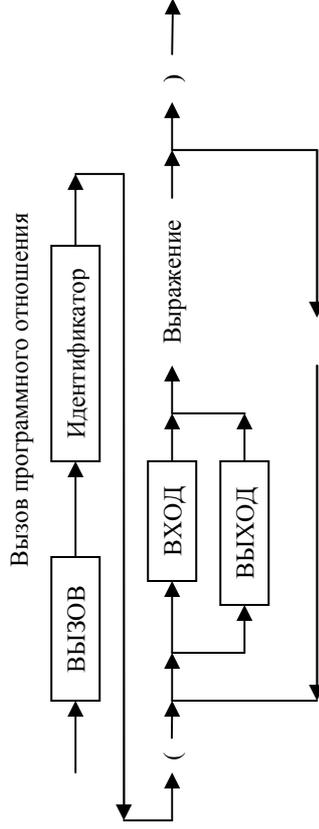
ГДЕ (K >= 0 ; K <= 1 ;

$X = M.A.X * (1 - K) + M.V.X * K;$

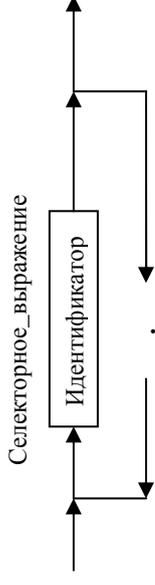
$Y = M.A.Y * (1 - K) + M.V.Y * K ;$

Использование кванторов – это введение дополнительных (связанных) переменных в описание модели. Проверка значений логических выражений, задаваемых кванторами всеобщности, осуществляется последовательным перебором всех элементов рассматриваемого в кванторе множества. Поэтому для реализуемости такой проверки необходимо, чтобы это множество было конечным.

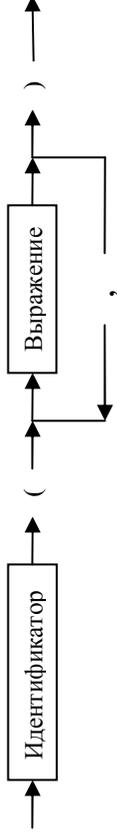
В описании множеств допустимо использование программных отношений, реализованных с помощью алгоритмических языков (фортран и т.п.). Эти отношения в языке имеют следующий вид:



Здесь идентификатор — имя соответствующей подпрограммы, а выражения — фактические параметры, снабжённые указанием на характер их использования.



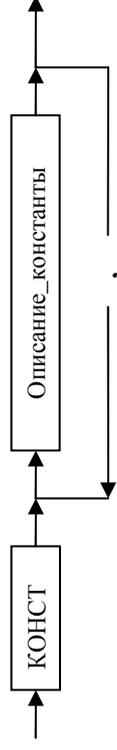
Стандартная функция



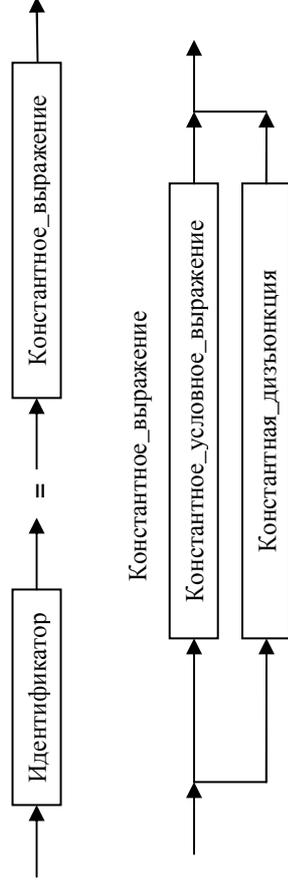
В качестве стандартных функций могут быть использованы обычные, встроенные в алгоритмические языки функции (sin, cos, log и т.д.).

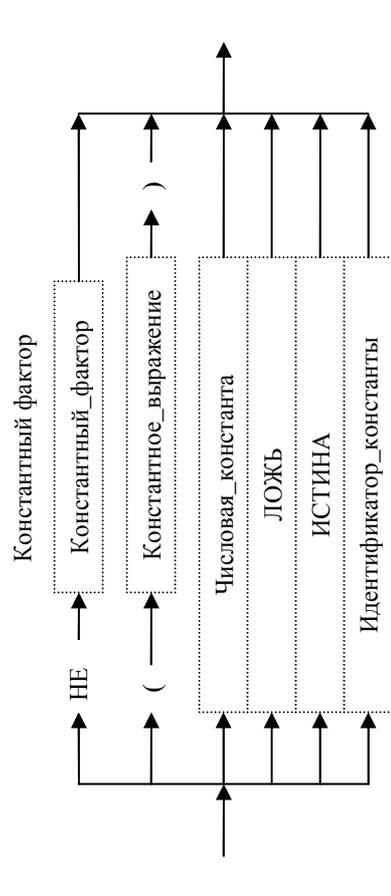
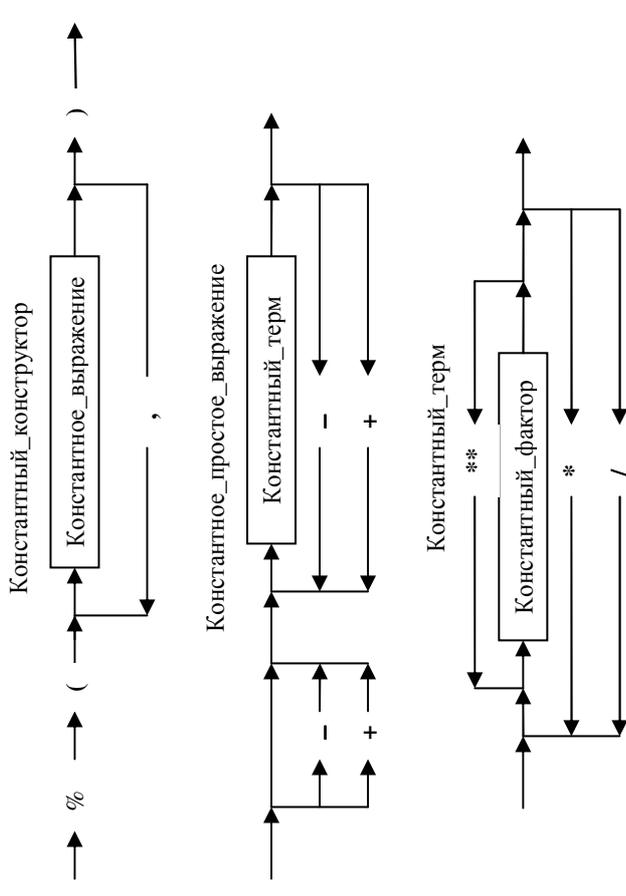
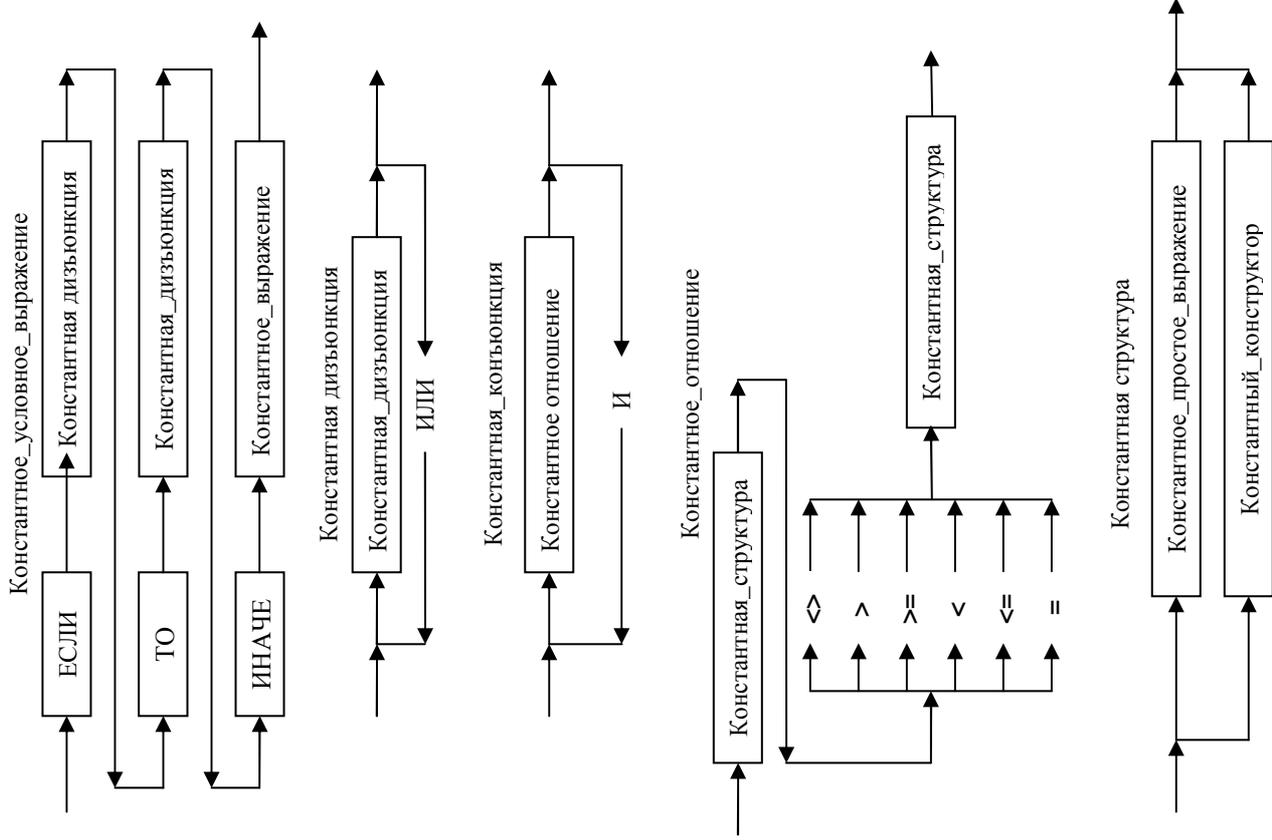
В языке существует возможность мнемонического задания постоянных числовых и векторных значений для их последующего использования при описании множеств и других констант.

Описание констант

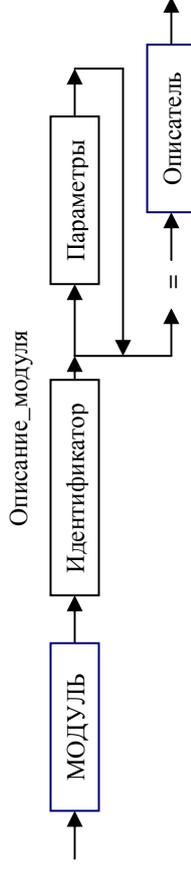


Описание константы





Описание моделей с помощью множеств преследует своей целью конструкторивное вычисление элементов этих множеств. Необходимость конструкторивного вычисления требует, во-первых, конечности этого множества и, во-вторых, специального указания, что такое вычисление должно быть произведено. Для этого служит спецификация модулей.



Построенные с помощью представленного здесь языка модели преобразуются в специальные внутренние структуры или в готовые программные модули и могут образовывать прикладные библиотеки, входящие в состав баз знаний, использоваться автономно и т.д. До сих пор речь шла о построении вычислительных моделей и программ, в то время как постановка задач проектирования обязательно включает в себя описание формы представления входной и выходной информации.

Описание входной и выходной информации

Одним из возможных подходов реализации информационного обмена пользователя с программами является организация ввода-вывода данных (ВВД) через специально созданные электронные бланки (ЭБ). Электронный бланк с точки зрения пользователя — это активная часть экрана (весь экран терминала, которая содержит некие "поля", служащие для ввода-вывода данных, а также вспомогательную информацию, не подлежащую изменению в процессе ввода-вывода. ЭБ должны поддерживать все необходимые функции по редактированию полей, а также обеспечивать контроль таких параметров вводимой информации, как типы данных, их размерности и, возможно, принадлежность числовых данных некоторым диапазонам. Кроме того, ЭБ обеспечивают недоступность служебной информации для редактирования пользователем. В состав комплекса ВВД входят средства, позволяющие отдавать ЭБ двумя способами: с помощью специализированного редактора ЭБ, а также путём описания ЭБ на некотором языке. Описания ЭБ, сделанные одним из

указанных способов, транслируются в единое представление, в котором они доступны для вызова из программ, написанных на традиционных языках программирования, а также могут быть составной частью описания задач и объектов.

Отметим, что требования к набору ЭБ по составу и спецификациям формируются разработчиками САПР на этапе постановки задачи.

Редактор ЭБ — программа из состава комплекса ВВД, позволяющая формировать на экране электронный бланк и записать в файл некоторое его внутреннее описание, соответствующее содержимому экрана. Работая непосредственно в редакторе ЭБ, программист формирует поля для ввода-вывода данных, размещает их на экране, снабжает электронный бланк служебной информацией (возможно, графической) для пользователя. Очевидно, что, помимо описания внешнего представления ЭБ на экране, полное описание ЭБ может содержать дополнительную информацию. Эта информация описывает типы полей (типы данных, вводимых через поле), такие атрибуты полей, как, например: начальное содержимое поля, положение десятичной точки, направление заполнения поля (от первого символа вправо или от последнего символа влево), номер шрифта или регистр клавиатуры для символьных полей и т.д.

В состав описания ЭБ входит также и информация о некоторых атрибутах собственно электронного бланка. Это — информация о размерах ЭБ, о позиционировании его на экране дисплея, о взаимодействии с другими ЭБ и т.д.

Как было сказано выше, одним из способов формирования ЭБ может быть описание его на специализированном языке. Очевидно, что конструкции этого языка должны позволять описывать ЭБ и его особенности в полном объёме и соответствовать в этом смысле функциям редактора электронных бланков.

Приведём пример возможного описания электронного бланка на некотором специализированном языке.

```

Эл. бланк..... БЛОК_ВВОДА_ДАТЫ
  строк ..... 10
  колонок ..... 50
  поз. стр. .... 5
  поз. кол. .... 30
  ...
Поле ..... ЧИСЛО
  тип ..... ЦЕЛОЕ
  поз. стр. .... 2
  поз. кол. .... 15
  ...
Поле ..... МЕСЯЦ
  тип ..... СИМВ20
  поз. стр. .... 3
  поз. кол. .... 15
  
```

С точки зрения архитектуры САПР комплекс ВВД является промежуточным звеном между пользователем и используемыми в системе моделями. Выше был рассмотрен возможный механизм обмена данными между пользователем и электронными бланками (их информационными буферами). Этот механизм инвариантен по отношению к вычислительным моделям, используемым в системе. Для обеспечения соответствия между полями ЭБ и переменными в конкретных вычислительных моделях в состав комплекса ВВД входит пакет специализированных процедур. Он используется программистом при написании подсистем, использующих ввод-вывод данных через ЭБ.

В первую очередь это процедуры, обеспечивающие вызов из программы конкретного электронного бланка и его включение в работу. Также в состав пакета входят процедуры, осуществляющие присваивание переменным моделей значений соответствующих полей ЭБ; при этом необходим контроль соответствия типов переменных модели и содержимого полей (возможен также семантический анализ содержимого поля с последующим повторением ввода). В отдельную группу выделяются процедуры вывода значений переменных модели в поля электронных бланков.

Программы, используемые для организации ввода-вывода электронных бланки, настраиваются на определённые наборы ЭБ, организованные в библиотеки. Специальные процедуры из состава комплекса ВВД осуществляют подключение поименованных электронных бланков и обеспечивают обмен между информационными буферами ЭБ и программными переменными.

Функционирование электронных бланков в период редактирования их пользователем системы обеспечивается некой резидентной программой (РП) из состава комплекса ВВД. РП позволяет пользователю осуществлять такие операции, как редактирование полей, переход от одного поля к другому, просмотр выходной информации (текстовой или числовой). Важнейшей функцией РП является контроль корректности действий пользователя. Такие ошибки пользователя, как ввод недопустимого символа в поле, попытка переполнения поля, поиск несуществующего поля и т.д., идентифицируются резидентной программой, которая выдаёт пользователю сообщение о его некорректных действиях и, если это необходимо, обеспечивает повторный сеанс редактирования всего электронного бланка или отдельных его полей.

Отметим, что при такой организации ввода-вывода данных ЭБ не связаны жёстко с вызывающими их программами, поскольку их функционирование обеспечивается резидентной программой комплекса ВВД, т.е библиотеки

ЭБ взаимодействуют с пользовательскими загрузочными модулями через посредство РП. Таким образом, изменение наполнения библиотек ЭБ, а также редактирование отдельных ЭБ не влечёт за собой реорганизации всей системы.

В современных САПР важную роль играет графическая информация, причём её наглядность и польза возрастают в сочетании с информацией других типов (текстовой и цифровой). Это означает, что применяемые в инструментальных средствах САПР графические системы (а это, как правило, системы ввода-вывода графической информации) должны бесконфликтно взаимодействовать с комплексом ВВД. Проблема в первую очередь состоит в том, что зачастую приходится использовать экран одного дисплея для ввода-вывода информации всех типов одновременно (например, через ЭБ производится ввод числовых данных и одновременно на экран осуществляется вывод графика функциональной зависимости), и информация различных типов может терять наглядность из-за наложений, кроме того, возможны сбои систем ввода-вывода.

Для решения этой проблемы при выборе конкретных систем ввода-вывода необходимо учитывать способы генерации изображения на экране. Желательно, чтобы для систем графического и алфавитно-цифрового ввода-вывода эти способы были различны. Заметим, что эти вопросы существенно зависят от конструкции и математического обеспечения используемых рабочих станций (дисплеев).

Для повышения наглядности и читаемости информации на экране может быть введена некоторая дисциплина, регламентирующая взаимное расположение информации различных типов, например, введение специальных "окон" на экране для размещения только графической информации и т.д.

Литература

1. Вышинский Л.Л. Структура моделей в задачах проектирования. Наст. сб.
2. Вышинский Л.Л., Прибытков Ю.Д., Шиленко В.И., Широков Н.И. Инструментальная система ФАКИР // Изв. АН СССР. Сер. Техн. киберн. 1986. № 3. С. 35-40.
3. Гринёв И.Л., Широков Н.И. Системы управления базами данных в САПР. Наст. сб.

СРЕДСТВА УПРАВЛЕНИЯ ДАННЫМИ В САПР

И.Л. Гринёв, Н.И. Широков

Введение

Создание САПР немислимо без использования сложных структур данных, применяемых для представления в памяти ЭВМ моделей объектов проектирования. Многолетняя практика программирования выработала большое количество разнообразных методов организации сложных информационных структур управления памятью. В этой области произошёл качественный скачок, выразившийся в создании концепции баз данных как метода формальной спецификации хранимой в ЭВМ информации. В литературе описаны три типа систем управления базами данных (СУБД): иерархические, сетевые и реляционные.

Иерархический подход можно рассматривать, с одной стороны, как результат развития и совершенствования методов доступа в файловых системах, а с другой стороны, как первый шаг на пути к современным сложным СУБД. В настоящее время СУБД такого типа представляют скорее исторический интерес.

Сетевой подход, сформулированный комитетом КОДАСИЛ [1], представляет собой обобщение многочисленных методов организации сложных структур данных и основывается на использовании отношения "один ко многим" для выражения взаимосвязей между фрагментами хранимой информации.

Дальнейшим развитием методов спецификации данных стал реляционный подход — описание отношений в виде двумерных таблиц. Очевидное

преимущество этого подхода — концептуальная целостность и простота — сочетается со сложностью эффективной программной реализации.

Таким образом, перед разработчиком САПР встаёт нетривиальная задача выбора типа СУБД и конкретной системы с учётом её эксплуатационных характеристик. Так, сетевые СУБД характеризуются эффективным доступом к данным, но относительно сложным программированием запросов и сравнительно малой гибкостью в смысле модификации структуры хранимой информации.

Реляционные СУБД, в свою очередь, сочетают простоту программирования запросов и гибкость с относительно невысокими временными характеристиками выполнения запросов.

1. Сетевая СУБД ГЕНБД

В процессе реализации автоматизированной системы весовых расчётов АСВР [2] и программного комплекса для проектирования электросхем бортового оборудования ЭЛМОН [3] разрабатывалась СУБД ГЕНБД.

Ранние версии системы ГЕНБД реализовывались как инструментальные системы генерации СУБД по заданному описанию логической схемы базы данных. ГЕНБД-3, являющаяся их развитием, реализовывалась как совокупность следующих программных компонент:

- транслятор схем;
- генератор программы создания пустой базы данных;
- генератор подпрограммы манипулирования файлами;
- генератор подпрограмм монопольного интерфейса;
- генератор подпрограмм мультizaдачного интерфейса;
- генератор программ диалогового интерфейса;
- генератор программы диагностики и коррекции;
- транслятор подсем;

генератор подпрограмм — операторов языка манипулирования данными (ЯМД).

Исходной информацией для генерации СУБД являются: описание схемы базы данных, совокупность описаний подсхем. Схема базы данных является подробной спецификацией тех данных, которые будут храниться в БД. Эта спецификация включает в себя описание типов записей и типов наборов. Тип записи определяет характер хранимой информации, а тип набора — структурные связи между различными фрагментами данных. Подсхема — это описание интерфейса конкретной программы с СУБД. Использование подсхем даёт возможность ограничить для программ доступ к данным только заданными рамками. Ограничение доступа обеспечивается указанием списка объектов (типов записей, элементов записей и типов наборов) и списка операций над этими объектами. Язык описания схемы и язык описания подсхем в совокупности составляют язык описания данных (ЯОД).

Рассмотрим подробно функции каждой программы.

Транслятор схем предназначен для перевода исходного текстового описания схемы в её табличное представление, используемое при работе всех остальных программ. В процессе трансляции производится тщательный анализ описания схемы с целью выявления синтаксических и семантических ошибок. Генератор программы создания пустой базы данных предназначен для генерации текста фортран-программы, которая осуществляет запрос с терминала информации о максимальных размерах БД и создаёт совокупность файлов, в которых будут храниться данные, составляющие базу данных.

Генератор подпрограммы манипулирования файлами предназначен для генерации текста фортран-подпрограмм, реализующих непосредственную работу с файлами базы данных. Эти подпрограммы обрабатывают записи файлов в соответствии со специальным протоколом связи — интерфейсом микроуровня

работы с БД. Все остальные интерфейсы непосредственно выходят на уровень протокола связи.

Генератор подпрограмм монопольного интерфейса предназначен для генерации фортран-текста подпрограмм, реализующих протокол связи с БД в монопольном, однозадачном режиме. Монопольный режим означает возможность одновременной работы с данным экземпляром БД только одной задачи.

Генератор подпрограмм мультизадачного интерфейса предназначен для генерации фортран-текста подпрограмм, реализующих протокол связи с БД в мультизадачном режиме. В этом режиме допускается одновременная работа с БД нескольких задач. Система осуществляет синхронизацию запросов к БД в целях обеспечения целостности данных. Реализованы средства защиты от модификации данных, одновременно обрабатываемых в других задачах.

Генератор программ диалогового интерфейса предназначен для генерации фортран-текста интерпретатора команд диалогового языка манипулирования данными. Интерпретация команд производится посредством использования команд протокола связи.

Генератор программы диагностики и коррекции предназначен для генерации фортран-текста программы диагностики и коррекции. Указанная программа осуществляет тщательный анализ содержимого файлов БД на предмет непротиворечивости и целостности информации. В случае обнаружения ошибок в данных производится их коррекция.

Транслятор подсхем предназначен для перевода исходного текстового описания подсхемы в её табличное представление, используемое в генераторе подпрограмм языка манипулирования данными. В процессе трансляции производится тщательный анализ описания подсхемы с целью проверки её соответствия схеме и выявления ошибок.

Генератор подпрограмм языка манипулирования данными предназначены для генерации фортран-текста совокупности подпрограмм и файла описания общих переменных, обеспечивающих реализацию языка манипулирования данными операторами фортрана-77.

Рассмотрим теперь язык манипулирования данными. Обработка информации посредством ЯМД основывается на использовании рабочей области. Рабочая область — это совокупность переменных, соответствующих элементам схемы. Каждому элементу схемы в рабочей области соответствует переменная, идентификатор которой совпадает с именем элемента. Тип и размер переменной соответствуют описанию элемента. В фортранном диалекте ЯМД переменные рабочей области сгруппированы в общие блоки и соответственно доступны как пользователю — программисту, так и СУБД. В диалоговом диалекте доступ к переменным осуществляется дополнениями к командам. При записи информации в БД значение элементов вначале помещается пользователем в рабочую область, а оттуда СУБД переписывает их в БД. При считывании информации из БД значения элементов извлекаются СУБД в рабочую область, откуда пользователь может их брать для дальнейшей обработки.

В процессе работы программ с БД СУБД хранит для каждого типа записи и набора по одному текущему экземпляру. Текущие экземпляры называются индикаторами состояния. Выполнение операторов ЯМД заключается в изменении значений индикаторов состояния (содержимого и связей друг с другом соответствующих экземпляров записей). Имеются две основные группы операторов ЯМД: операторы работы с записями и операторы работы с наборами. В операторах работы с записями указывается тип записи, а в операторах работы с наборами — тип набора.

Операторы ЯМД — запросы к базе данных — могут использоваться из программы диалогового интерфейса и из программ пользователя, написанных

на языках высокого уровня (реализация ЯМД как процедурного расширения основного языка).

Последняя версия ГЕНБД (ГЕНБД-4) реализована в виде совокупности программных компонент без использования методов генерации. В состав программных компонент входит библиотека объектов модулей, реализующих операторы ЯМД, и совокупность утилит для ведения баз данных:

утилита создания базы данных;

утилита реформатирования базы данных из формата ГЕНБД-3;

утилита диалогового интерфейса;

утилита диагностики и коррекции;

утилита управления мультирежимом;

утилита сортировки и балансировки наборов;

утилита компрессирования базы данных;

утилита слияния баз данных;

утилита реструктуризации базы данных.

Реализация СУБД с использованием генерации её программных компонент позволяет существенно упростить проблемы управления памятью для; предоставления логической схемы базы данных, а также повысить временную эффективность операторов ЯМД (учёт конкретных особенностей схемы на этапе генерации). Реализация СУБД вторым способом даёт большую гибкость разработчику САПР, так как описание подсхемы (спецификация интерфейса программы с базой данных) осуществляется средствами ЯМД, а не ЯОД. В связи с этим синтаксис ЯМД претерпел соответствующие изменения.

2. Проблемы вариантного хранения информации

При автоматизированном проектировании возникает необходимость хранения в базе данных нескольких вариантов одного проекта, имеющих

незначительные отличия. Из соображений поддержания целостности информации целесообразно дублирование повторяющихся её фрагментов. В ходе создания СУБД ГЕНБД-4 была разработана методика организации и ведения вариантных баз данных, представляющая собой расширение сетевого подхода.

Предлагаемая методика пригодна для создания вариантных СУБД сетевой структуры, в которых логические связи реализуются ссылочными полями.

Основные понятия: экземпляр записи, содержательное поле записи, ссылочное поле записи. Операторы языка манипулирования данными — алгоритмы, составленные из следующих микроопераций:

создание нового экземпляра записи с присваиванием пустых значений всем его ссылочным полям;

уничтожение экземпляра записи, если все его ссылочные поля имеют пустые значения;

выбор экземпляра записи по значению ссылочного поля другого экземпляра (переход по ссылке);

модификация значения ссылочного поля заданного экземпляра записи; модификация значения содержательного поля заданного экземпляра записи.

Если у нас есть СУБД, построенная на изложенных принципах, то её можно обобщить до вариантной СУБД. Основное понятие вариантной БД — множество вариантов. Все множества вариантов образуют древовидную структуру по отношению к теоретико-множественной операции включения. Если v_2 есть потомок v_1 в дереве вариантов, то v_2 является подмножеством v_1 . В противном случае v_1 и v_2 не пересекаются. Терминальные вершины дерева вариантов могут интерпретироваться как отдельные варианты, а не множества. Деревья вариантов допускает единственный способ модификации — расщепление терминального множества вариантов на совокупность пересекающихся подмножеств, что соответствует

одновременному подвешиванию новых вершин к терминальной вершине.

Вариантность отношений в вариантных СУБД реализуется заданием значений ссылочных полей как множеств альтернатив вида

$$V = \{(v_i, r_i) \mid i = 1, \dots, n\}, \quad (1)$$

где v_i — множество вариантов, r_i — соответствующее значение ссылки. Выполняется условие: v_i и v_j попарно не пересекаются и все r_i — не пустые ссылки.

Вводится новая микрооперация — выделение подварианта. Если заданы v_c — текущее множество вариантов и ссылочное поле в виде (1), то ищется v_i такое, что v_c является подмножеством v_i . Если найдено v_i , отличное от v_c , то v_i заменяется в V на множество его непосредственных потомков. Алгоритм повторяется до тех пор, пока не будет $v_c = v_i$. Множество v_i единственно по условию, наложенному на организацию ссылочного поля. Алгоритм не нарушает это условие в силу ограничения на организацию дерева вариантов.

Если фиксировано текущее подмножество вариантов v_c , то для данной вариантной БД определим БД, конкретизируя каждое ссылочное поле до значения:

$$r = \begin{cases} r_i, & \exists (v_i, r_i) \in V \ \& \ v_c \subseteq v_i, \\ \emptyset & \end{cases}$$

В силу свойства попарной непересекаемости v_i конкретизируемое значение определяется однозначно.

Любой оператор ЯМД выполняется для заданного текущего подмножества вариантов v_c . Применение оператора ЯМД должно сохранять смысл при конкретизации вариантной БД любым множеством вариантов W . Если W — подмножество v_c , то конкретизированная БД до выполнения оператора ЯМД и после выполнения отличается на результат применения этого оператора

к обычной (невариантной) БД. Если W не является подмножеством v_c , то W и v_c не пересекаются (по определению дерева вариантов), и конкретизированные БД до и после выполнения оператора совпадают, что интуитивно соответствует смыслу вариантыности (этот оператор не затрагивает другие варианты).

Алгоритмы операторов ЯМД строятся на соответствующих алгоритмах в невариантных БД с точностью до определения микроопераций.

Микрооперация БД создания экземпляра записи с присваиванием пустых значений ссылочным полям переносится в БД непосредственно.

Микрооперация БД уничтожения экземпляра записи переносится в БД следующим образом. Применяется выделение подварианта для всех ссылочных полей. Если в результате какое-либо v содержит пару (v_c, r_c) , то условие, что все ссылочные поля пусты, признаётся невыполненным. Иначе производится физическое уничтожение экземпляра, если все v пусты. Если какие-либо v не пусты, то указанный экземпляр сохраняется для других подмножеств вариантов.

Микрооперация БД выбора экземпляра записи по ссылочному полю другого экземпляра реализуется вычислением ссылки для заданного v_c .

Микрооперация БД модификации ссылочного поля реализуется в БД выделением подварианта и корректировкой пары (v_c, r_c) .

Микрооперация БД модификации значения содержательного поля данных может потребовать создания нового экземпляра записи после выделения подварианта (v_c, r_c) , если некоторые v содержат другие пары (v, r) . Наличие обратных ссылок может потребовать выделения подвариантов соседних экземпляров записей и их корректировки.

1. Язык описания данных КОДАСИЛ. М.: Статистика, 1981.
2. Скобелев С. И., Широков Н. И. Весовой анализ и контроль в САПР ЛА. Наст. сб.
3. Прибытков Ю. Д., Шиленко В. И. Программный комплекс ЭЛМОН. Наст. сб.

Исходными для расчёта ЛТХ являются уравнения динамики материальной точки. Внешние силы обычно представлены экспериментальными и расчётными зависимостями в табличной форме или параметрическими семействами кривых. В качестве альтернативы графоаналитическим методам, которые до недавнего времени использовались в инженерной практике, в этом случае следует рассматривать такие методы, которые были бы просты и надёжны при численном исследовании уравнений, правые части которых не задаются аналитическими функциями. Анализ исходной и расчётной информации (полётные условия, представление внешних сил и характеристик), необходимый при разработке логических структур данных, приводит к следующему двум типам данных: параметры и зависимости.

Под термином параметризация расчётов подразумевается выполнение многократной серии вычислений, в которых ряд исходных параметров варьируется в заданных диапазонах. Результирующие данные в этом случае можно представить в виде обыкновенных таблиц или в форме табличных функций, описывающих требуемые зависимости. Последние могут быть использованы при номографировании ЛТХ.

Проблема, связанная с хранением и получением доступа к различному рода числовым таблицам, описывающим функциональные зависимости, заключается в разработке программных средств, обеспечивающих единообразный подход при передаче информации подобного рода между различными подсистемами и САПР.

Логическая структура комплекса ЛТХ включает интерпретатор команд, рабочую область, блок расчётных модулей, библиотеки табличных функций.

Интерпретатор команд представлен группой подпрограмм, предназначенных для ввода и синтаксического анализа командных строк. Результат

ОСНОВЫ ПОСТРОЕНИЯ ПРОГРАММНОГО КОМПЛЕКСА ЛЁТНО-ТЕХНИЧЕСКИХ ХАРАКТЕРИСТИК В САПР ЛА

В.Л. Катунин

Вычислительные и программные средства, применяемые или разрабатываемые при проектировании ЛА, включают различные подсистемы, например, ПРОЧНОСТЬ, АЭРОДИНАМИКА, ГЕОМЕТРИЯ и др. Понятие подсистема САПР подразумевает широкий круг вопросов и задач в предметной области, начиная от программной реализации расчётных алгоритмов, поддержки архивов, обеспечения диалога и кончая управлением вычислительными процессами.

Множество задач, связанных с определением лётно-технических характеристик самолёта (ЛТХ), включает различные аспекты, с которыми приходится сталкиваться практически на всех стадиях проектирования. Например, на этапе формирования облика речь идёт о выборе конструктивной схемы, удовлетворяющей заданным тактико-техническим требованиям (ТТ), в число которых входит дальность или продолжительность полёта, максимальная скорость, перегрузка и др. характеристики. При детальной проработке проекта в результате расчёта ЛТХ определяются условия функционирования отдельных агрегатов самолёта. При проведении лётных испытаний важной проблемой является идентификация расчётных данных с результатами натурного эксперимента. Техническая документация, выпускаемая при сдаче самолёта в эксплуатацию, содержит разнообразные графические зависимости (номограммы), представляющие в полном объёме области применения.

работы интерпретатора определяется типом заданной команды. В случае некорректного задания выдается сообщение об ошибке и управление передается на повторный ввод.

Рабочая область — это часть операционной памяти, в которой хранятся данные, доступные для всех программ комплекса. Рабочая область связывает комплекс в единое целое и обеспечивает сообщение между различными программами единицами. Здесь содержится информация о параметрах комплекса, исходных и расчётных зависимостях, выходных и входных списках. Сохраняемое рабочей области можно сохранять в архивных файлах, а затем восстанавливать с помощью некоторых дополнительных команд. Команды задания значений параметров или ввода исходных зависимостей изменяют содержание рабочей области.

Блок расчётных модулей включает программные компоненты, каждая из которых представлена последовательностью подпрограмм, составляющих вычислительное ядро комплекса. Это множество подпрограмм можно разбить на несколько групп, в соответствии с характером выполняемых операций: расчёт аэродинамических коэффициентов, высотно-скоростных характеристик двигателя, составляющих перегрузки; контроль и обработка параметров, представляющих ограничения или условия для траектории полёта; форматирование и табулирование результатов расчёта. Передача управления в блок расчётных модулей осуществляется в том случае, когда на вход в интерпретатор поступает расчётная команда. Каждой команде этого типа соответствует определённый модуль блока.

Библиотеки табличных функций обеспечивают хранение и доступ к разнообразным исходным и расчётным зависимостям, которые представлены в виде числовых таблиц. Библиотека реализована в виде файла, который содержит имена функций, даты занесения, число переменных, значения переменных

в узлах сетки разбиения. Средства, обеспечивающие интерфейс между ; библиотеками и прикладными программами, позволяют копировать содержимое библиотеки в рабочую область, вычислять значения функции, создавать и сохранять расчётные зависимости.

Множество команд комплекса можно разделить в соответствии с функциональным назначением на несколько типов.

Команды 1-го типа имеют формат

<НАЗВАНИЕ ПАРАМЕТРА> = <СПИСОК ЗНАЧЕНИЙ>

и предназначены для задания значений рабочих параметров комплекса. Название команды совпадает с общепринятым именем полётного или управляющего параметра, например, Н – высота, ОР – режим работы двигателя, М – число Маха. Значения, заданные одним из следующих способов:

0 2000 5000 11000 (для высоты)

или

0.4 : 1.0 0.05 (для числа Маха),

используются при выполнении параметрических расчётов. При выполнении простых расчётов выходные величины определяются для текущих значений исходных параметров. Ряд параметров комплекса предназначен для задания* разнообразных расчётных условий. Например, команда

PTERM/TIME = 15.2

определяет в качестве параметра окончания расчёта траектории (PTERM), продолжительность полёта (TIME) и устанавливает требуемую величину. Формат команд задания условий отличается от рассматриваемого, так как требует указать вспомогательный параметр. Таблицы 1, 2 содержат описание некоторых команд рассматриваемого типа.

Таблица 1

Имя параметра	Размерность	Интервал допустимых значений	Описание
M	–	(MMIN, MMAX)	Полетное число Маха
H	м	(HMIN, HMAX)	Высота
ALPHA	град.	(ALMIN, ALMAX)	Угол атаки
CY	–	(-9.99, 99.99)	Коэффициент подъёмной силы
NX	–	(-9.99, 99.99)	Осевая составляющая перегрузки
Q	кг/м/сек/сек	(QMIN, QMAX)	Скоростной напор
U	км/час	(UMIN, UMAX)	Полетная скорость
DR	–	(DRMIN, DRMAX)	Работа двигателя

Таблица 2

Название параметра	Список вспомогательных параметров	Назначение
PZOOM	M, V, VDVC, HE, Q, THETA	Задание условий набора высоты или "планирования"
PVOT	MX, VMX	Задание условий определения области установившегося полёта
PMACH	M, V, VDVC	Задание величины скорости
PORT	FUEL, TIME, WAY, NY, IMEGA	Задание критерия оптимизации
PTERM	Любые параметры	Задание условия окончания расчёта траекторных задач
PTURN	NY, GAMMA, RADIUS	Задание условия пространственного разворота

Команды 2-го типа имеют формат

<НАЗВАНИЕ ЗАВИСИМОСТИ = <МОДУЛЬ> <БИБЛИОТЕКА>

и служат для ввода исходных числовых зависимостей, например, аэродинамических коэффициентов, высотно-скоростных характеристик двигателя и др. Аргумент команды указывает местоположение задаваемой функции.

Таблица 3

Название	Список переменных	Описание
CXIF	MESH, M, CY	Индуктивная составляющая коэффициента лобового сопротивления
CXADDF	NADD, M	Поправка к коэффициенту лобового сопротивления за счёт дополнительных подвесок
CYF	MESH, M, ALPHA	Коэффициент подъёмной силы
QSECF	DR, H, M	Секундный расход топлива двигателя
MMAXF	H	Верхняя граница чисел Маха по высоте
DMDVCF	M	Поправка к величине "приборной" скорости

Команды, предназначенные для формирования входа-выхода, имеют формат

(OVT / РАСЧЕТНАЯ КОМАНДА) = <СПИСОК ПАРАМЕТРОВ>

INP = <СПИСОК ПАРАМЕТРОВ>

TAB = <СПИСОК ПАРАМЕТРОВ>

STER = <ПАРАМЕТР>

С помощью команды OVT задаётся список выходных параметров при выполнении расчётов. <РАСЧЕТНАЯ КОМАНДА> указывает название расчёта, результаты которого содержат лишь значения параметров выходного списка.

Команду INP можно использовать для задания входного списка при выполнении параметрических расчётов. В этом случае выходные величины

определяются для всех значений параметров списка, тогда как остальные входные параметры не изменяются.

Команда INP с пустым списком устанавливает режим непараметрических расчётов.

Команда TAB предназначена для формирования зависимостей расчётных величин от входных параметров, заданных командой INP. Результаты в этом случае представляют собой табличные функции.

Команда STEP предназначена для задания параметра табулирования при выполнении расчётов траекторных задач. В этом случае результаты расчёта выдаются в те моменты, когда указанный параметр принимает заданные значения.

Следующие примеры демонстрируют действие отдельных команд расматриваемого типа.

1. Результаты параметрического расчёта осевой перегрузки:

```
FLY> OUT/FNX = NX, CX, CY, ALPHA
FLY> PTURN/NY = 2 4 6
FLY> INP = PTURN
FLY> FNX
```

NX	CX	CY	ALPHA	PTURN
0.5111	0.0702	0.232	4.6	2.00
0.0273	0.1291	0.465	9.3	4.00
-0.8695	0.2381	0.697	13.9	6.00

```
FLY>
```

2. Результаты табулирования зависимости осевой перегрузки по весу и режиму работы двигателя:

```
FLY> INP = W, DR
FLY> W = 10000 15000 20000
FLY> DR = 1:2 0.5
FLY> TAB = NX
FLY> FNX
```

3

3

1.00	1.50	2.00
10000.	15000.	20000.
0.4890	0.3047	0.2043
0.9117	0.5865	0.4157
1.3344	0.8683	0.6270

Совокупность расчётных команд разбита на три группы. К первой группе отнесены команды, которые можно использовать для контроля исходных зависимостей. Команды второй и третьей группы предназначены для выполнения расчётов локальных характеристик и решения траекторных задач. В таблице 4 приводится краткое описание некоторых расчётных команд.

Дополнительные команды предназначены для выполнения следующих операций: протоколирование в интерактивном режиме, сохранение и восстановление заданных значений параметров и функций, получение информации о параметрах, интерактивное или пакетное выполнение набора команд, сохранение табличных функций в библиотеках. Таблица 5 содержит перечень основных команд этого типа.

Фактическое определение максимальной дальности или радиуса действия в случае сложной траектории приводит к рассмотрению полётных профилей. Для большей части траектории заданы полётные режимы — полёт на малой высоте, набор высоты при заданных условиях и др. Профиль полёта состоит из типовых участков, расчёт каждого из которых в отдельности достаточно прост.

Таблица 4

<i>Команда</i>	<i>Назначение</i>
ATM	Вычисление параметров стандартной атмосферы
CLIMB	Расчёт траектории полёта в вертикальной плоскости Набор высоты
CRUISE	Расчёт характеристик крейсерского полёта
FCX	Вычисление коэффициента лобового сопротивления
FCYNY	Вычисление коэффициента подъёмной силы по величине нормальной перегрузки
FDRSL	Вычисление высотно-скоростных характеристик двигателя
FMCY	Вычисление нижней границы диапазона чисел Маха по максимальной величине коэффициента подъёмной силы
FNX	Вычисление осевой перегрузки горизонтального полёта
FHMAX	Определение верхней границы допустимой области высот
FDRM	Расчёт оптимальной крейсерской скорости
FVY	Расчёт характеристик скороподъёмности

Общий формат последовательности команд, предназначенной для формирования и расчёта профилей в комплексе ЛТХ, выглядит следующим образом:

```

START
<НАЧАЛЬНЫЙ ПАРАМЕТР>
<НАЗВАНИЕ ПЕРВОГО УЧАСТКА>
<ПАРАМЕТР ДЛЯ ПЕРВОГО УЧАСТКА>
...
...
<НАЗВАНИЕ ПОСЛЕДНЕГО УЧАСТКА>
<ПАРАМЕТР ДЛЯ ПОСЛЕДНЕГО УЧАСТКА>
FINISH

```

При задании текущего участка следует указать его название и некоторые параметры, определяющие полётные условия. Данные об участках задаются последовательно, в том порядке, в котором они составляют профиль. Ниже приводится описание отдельных участков.

После послышки команды START необходимо ввести начальные значения ряда параметров (веса, высоты, скорости) и данные о сбрасываемых топливных баках, которые представлены запасом топлива, весом и показателем лобового сопротивления.

Участок крейсерского полёта задаётся командой CRUISE. В число параметров, определяющих условия расчёта, входят крейсерская скорость или число Маха, параметр конечных условий, параметр оптимизации. Если отсутствует параметр конечных условий, то данный участок считается неопределённым крейсерским участком. Если профиль содержит один подобный участок, то речь идёт о расчёте максимальной дальности. Требуемый крейсерский расход топлива определяется с помощью несложной итерационной процедуры. Если задано два неопределённых участка, то выполняется расчёт радиуса действия. Крейсерские расходы топлива в этом случае определяются решением уравнений относительно величины расхода топлива и дальности на

Таблица 5

<i>Команда</i>	<i>Назначение</i>
<ИМЯ ФАЙЛА >	Выполнение последовательности команд в интерактивном режиме
EXAM <СПИСОК ПАРАМЕТРОВ>	Вывод значений полётных и управляющих параметров
EXIT	Окончание работы в комплексе
INSERT <БИБЛИОТЕКА> <ПАРАМЕТР>	Занесение полученной табличной функции в библиотеку
OPEN <ИМЯ ФАЙЛА>	Открытие протокольного файла
START	Формирование полётного профиля
STORE <ИМЯ ФАЙЛА>	Сохранение текущего состояния комплекса
SUB <ИМЯ ФАЙЛА>	Выполнение последовательности команд в пакетном режиме

неопределённых участках. Если задан параметр оптимизации, то крейсерская скорость определяется в соответствии с вспомогательным параметром для RORT.

Участок неустановившегося горизонтального полёта, для обозначения которого служит команда DYNAMO, можно использовать в качестве стыковочного для соседних участков профиля. В этом случае можно не задавать параметры, которые определяют условия расчёта, т.е. режим работы двигателя, параметр конечных усилий. Незвестные величины будут определены при расчётах предыдущего и последующего участков.

Для задания участков неустановившегося полёта в вертикальной плоскости (набор высоты) планирование, служат команды CLLIMB и GLIDE. В этом случае следует указать все необходимые параметры: режим работы двигателя, параметры конечных условий и условий полёта.

Участок "сброса груза" служит для изменения коэффициента лобового сопротивления и веса в результате сброса груза в определённых точках на траектории профиля. При выполнении расчёта радиуса действия этот участок используется в качестве граничного между последовательностями участков профиля в прямом и обратном направлениях.

Команда FINISH завершает ввод составляющих участков профиля, после чего выполняются некоторые проверки и, если всё нормально, выполняется расчёт.

ВЕСОВОЙ АНАЛИЗ И КОНТРОЛЬ В САПР ЛА

С.И. Скобелев, Н.И. Широков

Специфика весового проектирования летательных аппаратов (ЛА) состоит в необходимости достаточно точно учитывать массу большого количества различных агрегатов и бортовых систем, а также динамику их изменения в процессе создания ЛА. Проведение в процессе проектирования большого количества весовых расчётов и осуществление постоянного весового контроля требуют высокой степени автоматизации.

1. Весовое проектирование ЛА

Автоматизированная система весовых расчётов (АСВР) предназначена для решения следующих задач весового проектирования:

весового контроля;

распределения масс по элементам разбивки объекта;

весового анализа (сбор статистического материала).

АСВР позволяет синтезировать в памяти ЭВМ весовую модель проектируемого изделия и осуществлять её анализ.

Весовой контроль в конструкторском бюро представляет собой комплекс мероприятий, направленный на создание конструкции минимальной массы. Необходимым условием эффективного весового контроля является возможность получения оперативной информации о текущей массе изделия и любой его части.

Весовая информация представляется в любой момент времени по запросу пользователя в виде двух установленных форм:

Тип записи **OBJECT** – это множество изделий, для которых хранится весовая модель.

Тип записи **BASIS** вводит множество систем координат, используемых для задания весовой модели. Поля записи задают матрицу пересчёта локальных и глобальных координат. Матрица пересчёта определяется по векторам сдвига и углам поворота вокруг осей для данной системы координат по отношению к вышестоящей в иерархии.

Тип записи **BODY** – это множество узлов дерева конструкции. В полях записи задаются: имя узла, координаты центра масс узла, исходная информация для вычисления моментов инерции узла, значения лимитной, теоретической, чертёжной и фактической масс узла.

Тип записи **SECT** – это множество пространственных отсеков, на которые разбивается рассчитываемая конструкция.

Тип записи **PLANE** – это множество записей для хранения информации о положении отсека. Для прямоугольного отсека хранятся максимальные и минимальные значения координат. Для непрямоугольного задаётся список ограничивающих плоскостей. Каждая плоскость задана в виде трёх точек с учётом направления обхода.

Тип записи **NODE** вводит множество подсистем и сосредоточенных масс.

Набор **LOBJECT** обеспечивает доступ к списку изделий, хранимых в базе данных. Ключевая организация набора обеспечивает поиск изделия по имени.

Набор **LBASIS** задаёт список всех систем координат данного изделия и обеспечивает поиск системы координат по имени.

Набор **LBODY** задаёт список всех узлов дерева конструкции данного изделия и обеспечивает поиск узла по имени.

Набор **HBASIS** используется для задания иерархии систем координат. Величины сдвига начала координат и углы поворота осей для каждой системы, кроме корневой, заданы по отношению к вышестоящей.

Набор **HBODY** используется для задания структуры дерева конструкции. Этот набор задаёт конструктивное включение агрегатов, узлов, деталей, т.е. определяет отношение "состоит из".

Набор **COOR** определяет систему координат, в которой задан центр масс рассматриваемого узла.

Набор **LSECT** задаёт список всех отсеков данного изделия и обеспечивает поиск отсека по имени.

Набор **LPLANE** задаёт связь между отсеком и записями типа **PLANE** (информация о положении отсека).

Набор **SBODY** определяет факт принадлежности узла отсеку.

Набор **BSECT** определяет систему координат, используемую для задания информации о положении отсека.

Набор **LNODE** задаёт список подсистем и сосредоточенных масс.

Набор **LINK** задаёт последовательность узлов, которые принадлежат списку подсистем или сосредоточенных масс.

4. Архитектура системы

АСВР реализована в виде диалогового монитора и совокупности служебных программ. Пользователи системы работают с диалоговым монитором. В мониторе предусмотрена совокупность меню. Пользователь выбирает в текущем меню нужный раздел и отвечает на различные вопросы монитора в зависимости от выбранной команды. Монитор с помощью операторов языка манипулирования данными производит обработку хранимой в базе данных информации.

В мониторе предусмотрены следующие команды: просмотр списка изделий, выбор изделия для работы, создание нового изделия.

Если выбрано изделие, то предусмотрены следующие команды: работа с системами координат, с деревом конструкции с пространственными отсеками, с подсистемами и сосредоточенными массами, расчёт масс-инерционных характеристик.

При работе с системами координат предусмотрены команды: просмотр дерева систем координат, формирование дерева систем координат, корректировка системы координат, уничтожение системы координат, изменение структуры дерева систем координат.

При работе с деревом конструкции предусмотрены команды: просмотр дерева конструкции, формирование дерева конструкции, корректировка узла~дерева конструкции, уничтожение узлов дерева конструкции, изменение структуры дерева конструкции.

При работе с пространными отсеками предусмотрены команды: просмотр списка отсеков, создание нового отсека, корректировка отсека, уничтожение отсека.

При работе с подсистемами и сосредоточенными массами предусмотрены команды: просмотр списка подсистем, корректировка списка подсистем, просмотр списка сосредоточенных масс, корректировка списка сосредоточенных масс.

При расчёте масс-инерционных характеристик предусмотрены команды печати различных документов, отражающих текущее состояние весовой модели. Предусмотрен диалоговый и пакетный режимы счёта.

Служебные программы, входящие в состав АСВР, предназначены для администратора системы и предусматривают разнообразные действия, связанные с ведением базы данных, регистрацией пользователей и т.п.

5. Прикладные аспекты применения АСВР

Эксплуатация АСВР предусматривает заполнение специальных бланков конструкторами, выпускающими чертежи. В этих бланках задаётся фрагмент весовой модели агрегата с указанием наименований деталей (терминальных вершин дерева конструкции), сборочных единиц – агрегатов (нетерминальных вершин), а также информации о массе, координатах центра масс, моментом инерции, учётной информации. Заполненные бланки поступают в весовую

бригаду, где осуществляется контроль правильности их заполнения и ввод в ЭВМ, т.е. формирование весовой модели из отдельных фрагментов. Таким образом, в процессе проектирования в базе данных хранится весовая модель изделия, отражающая текущее состояние проекта, т.е. степень проработанности модели. По мере проработки конструкции в весовую модель вводятся новая информация, т.е. терминальные вершины дерева конструкции заменяются на соответствующие поддерева.

Наличие весовой модели, отражающей текущее состояние проекта, позволяет получать оперативные весовые сводки. Предусмотрено проведение расчётов на модели и выдача информации по нескольким установленным формам.

Выдача информации по форме 1 предусматривает поиск в базе данных максимальных поддереьев с заданными значениями классификационного признака (исполнитель, отдел, классификатор) и печать информации в виде, отражающем структуру дерева конструкции. В распечатываемой таблице приводятся массы соответствующих агрегатов и их соотношения с заданными лимитными значениями. Выдаются также итоговые данные для каждого значения классификационного признака. При этом массы нетерминальных вершин модели вычисляются как суммы соответствующих компонент – потомков в дереве конструкции. Предусмотрен учёт массы крепежа (заклёпок, винтов и т.п.) заданием суммарной массы крепежа для нетерминальной вершины с распределением этой массы по терминальным вершинам пропорционально их текущей массе. Информация по форме 1 позволяет оперативно отслеживать превышение лимитных значений массы для отдельных агрегатов, контролировать отдели-разработчики агрегатов и конкретных исполнителей.

Выдача информации по форме 2 предусматривает вычисление масс, координат центра масс и моментов инерции (собственных и полных, плоскостных и осевых), заданных агрегатов или изделия в целом с учётом разбиения агрегата на пространным ограниченными отсеки. Задаётся список отсеков

(хранящихся в базе данных), и производится просмотр терминальных вершин модели для заданного агрегата с проверкой принадлежности их к текущему отсеку. Для каждого отсека вычисляются суммарные значения массы вошедших в него деталей, координаты центра масс и значения моментов инерции. На печать выводятся таблицы имён деталей (терминальных вершин модели) и соответствующие значения вычисленных характеристик. Предусмотрена выдача итоговых таблиц по отсекам.

Разбиение изделия на отсеки необходимо для проведения прочностных расчётов с использованием метода конечных элементов. Масс-инерционные характеристики отсеков применяются для создания динамически подобных моделей, используемых при расчётах изделия на флаттер. При построении динамически подобных моделей возникает необходимость задания отдельных агрегатов в виде так называемых сосредоточенных масс, когда соответствующий агрегат не подвергается поэлементной проверке на принадлежность отсекам, а масс-инерционные характеристики вычисляются в виде суммы составляющих его деталей. Для этого в весовой модели предусмотрено задание соответствующих структур, выделяющих поддерева – сосредоточенные массы так, чтобы они не участвовали в расчётах отсеков. Расчёты по форме 2 предусматривают выдачу информации как по пустому изделию или его агрегату, так и по снаряжённому, т.е. с учётом деталей, входящих в дерево конструкции в виде отдельного, не связанного с заданным, поддерева. Например, требуется расчёт масс-инерционных характеристик крыла, в которое входят элементы (детали) системы электрооборудования, гидравлики и т.п. Для этого в весовой модели предусмотрены соответствующие структуры для выделения так называемых подсистем. Если корень некоторого поддерева весовой модели отмечен как подсистема, то просмотр входящих в неё деталей при размещении отсеков производится в процессе отдельного прохода по дереву подсистемы с тем, чтобы получить отдельные итоговые значения масс-инерционных характеристик соответствующей подсистемы.

Выдача информации по форме 3 предусматривает проведение масс-инерционных расчётов для заданного поддерева без учёта отсеков и выдачу информации, отражающей структуру дерева конструкции. Формат выдачи (состав столбцов таблицы значений отдельных масс-инерционных характеристик) задаётся пользователем.

В форме 2 и 3 предусмотрена выдача масс-инерционных характеристик в системе координат, заданной пользователем. При этом осуществляется соответствующий пересчёт характеристик, так как исходные данные задаются при формировании модели в различных (удобных для данного агрегата) системах координат. Системы координат организованы в древовидную структуру, входящую в состав весовой модели.

6. Выводы

Длительная эксплуатация АСВР позволила:

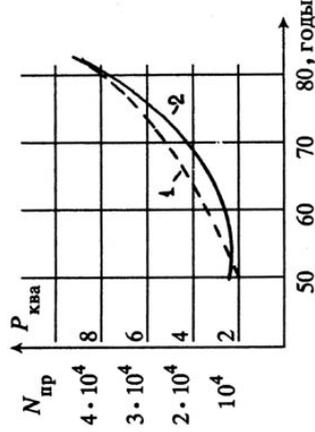
- повысить эффективность весового контроля;
- сократить сроки расчётов МИХ на 10-12 месяцев;
- сократить сроки изготовления и доработок динамически подобных моделей;
- корректировать проект до испытания и запуска в серию;
- повысить оперативность сбора и обработки статистических данных для весового анализа.

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ЭЛЕКТРОСХЕМ

М.Ю. Шапиро, В.И. Шиленко

Усложнение функций; увеличение объёма задач сложных технических объектов определяет широкое применение в них различных электротехнических и электронных устройств. Как следствие, возрастает объём и вес электрических схем объектов. В статье рассматривается подход к автоматизации проектирования электросхем – сложных технических объектов на примере летательных аппаратов.

Тенденция развития электротехнических и электронных устройств на ЛА среднего класса (с массой 20-30 тонн) приведена на рисунке.



Кривая 1 отражает рост мощности бортовых систем электроснабжения, кривая 2 — рост числа проводов на ЛА.

В начале 70-х годов с появлением бортовых вычислителей произошло резкое увеличение количества проводов на ЛА. Несмотря на предлагаемые

технические решения (мультиплексные связи, оптоволоконные системы), указанная тенденция сохраняется по настоящее время. Так, например, на самолёте В-1 фирмы North American внедрение мультиплексной системы позволило заменить около 10 тыс. проводов, т.е. 20% всех проводов. Тем не менее общее их количество увеличилось по сравнению с самолётом В-52 на 25-30% [1]. В связи с этим проектирование бортовых электросхем остаётся одной из наиболее трудоёмких и ответственных составляющих разработки ЛА, а её автоматизация существенно повышает производительность и качество работы проектировщиков ЛА.

Разработка бортового электрооборудования (БРЭО) может быть укрупнённо представлена следующим циклом проектных операций.

1. Рассмотреть ТЗ на ЛА и определить функциональные задачи оборудования.
2. Распределить функциональные задачи по системам и подсистемам оборудования.
3. Определить облик и структуру оборудования.
4. Разработать ТЗ на системы и подсистемы оборудования и задать их разработку (оформить договор и протоколы применения).
5. Разработать логику и протоколы взаимодействия систем и подсистем.
6. Определить поблочный состав.
7. Разработать сборочные и детальные чертежи блоков.
8. Разработать электросхемы блоков.
9. Разработать установочные чертежи блоков.
10. Разработать принципиальные электросхемы систем и подсистем.
11. Разработать схемы электрических соединений систем и подсистем.
12. Разработать алгоритмы и программы бортового оборудования.
13. Разработать методики контроля, проверки, поиска неисправностей.
14. Разработать эксплуатационно-техническую документацию.

15. Разработать методики наземных, лётных и других испытаний.
16. Откорректировать документацию по результатам испытаний и передать её в серийное производство.

Этот процесс носит итеративный характер, причём решения приходится принимать в условиях большого количества ограничений (по весу, габаритам, требованиям по электропотреблению, электромагнитной совместимости и т.д.).

Работы, указанные в пп. 1–16, с некоторой степенью условности можно привести к нормированным этапам разработки:

- эскизный проект (пп. 1–4);
- технический проект (пп. 5–6);
- рабочий проект (пп. 7–11);
- испытания (пп. 12–16).

Создание САПР, охватывающей все эти этапы, особенно стадии эскизного, технического проекта и испытаний, весьма проблематично из-за большого числа эвристических решений и плохо формализуемых процедур их принятия. В качестве первого шага целесообразно разработать САПР электросхем БРЭО, ориентированную на стадию рабочего проекта. Система должна удовлетворять ряду требований:

в связи с ориентацией на конструкторов, не проходящих специальную подготовку, необходимы развитые средства контроля за вводом/модификацией данных по электросхемам, включая выдачу рекомендаций в случае неверного ввода;

из-за существенных отличий в технологии проектирования, принятой в различных организациях, необходим развитый аппарат настройки на различные режимы функционирования, на семантику обозначений элементов схем, на их номенклатуру и вид графического отображения; фактически, конкретные знания о предметной области в системе должны размещаться в специальных настроечных файлах с возможностью их обновления администратором

системы без какой-либо перестройки программного обеспечения; возможность ограничения доступа к данным по схеме, устанавливаемого разработчиком схемы;

максимальное упрощение действий пользователя, предложение типовых данных в определённом контексте;

обеспечение построения графических документов в автоматическом режиме для наиболее трудоёмких и рутинных операций (определение габаритов элементов схем, зависящих от конкретной ситуации, изображение способов электромагнитной защиты проводов, компоновка элементов на листе и т.д.) и вместе с тем обеспечение "ручной" доработки листов в диалоговом режиме;

автоматическое построение комплекта текстовой конструкторской документации на основании данных, введённых пользователем;

возможность широкого использования проектных решений из прототипов; обеспечение комплекса мероприятий по постановке электросхем на контроль изменений и выпуску соответствующих документов.

Реализацией представления авторов о САПР БРЭО в качестве первого практического шага явилась система ЭЛМОН [2], разработанная в 1984 – 1986 гг. По результатам длительной промышленной эксплуатации в одном из ОКБ МАП получены следующие данные. Трудоёмкость выпуска формата АИ для конструкторской документации (КД) при разработке новых схем, не использующих прототипов, по традиционной технологии — 6 часов, с использованием систем ЭЛМОН – 4,8 часа. Для модифицируемой КД или выпуска преработанных извещений (при использовании прототипов) — 6 часов и 1,54 часа соответственно. При этом существенно снижается количество конструкторских ошибок, а ряд из них полностью ликвидирован, что сокращает время на доработку схем. Учитывая значительный объём КД, автоматизация проектирования и выпуска электросхем является весомым фактором повышения производительности труда конструкторов.

Опытная эксплуатация системы ЭЛМОН показала высокую надёжность и эффективность системы. Производительность труда конструкторов возрастает от 1,5–2 до 5–8 раз по сравнению с традиционными методами проектирования. Система внедрена в настоящее время в 6 организациях.

Литература

1. *Шатира М.Ю., Шиленко В.И.* Автоматизация проектирования электросхем. Наст. сб.
2. *Гринёв И.Л., Широков Н.И.* Средства управления данными в САПР. Наст. сб.

ПРОГРАММНЫЙ КОМПЛЕКС ЭЛМОН

Ю.Д. Прибытков, В.И. Шиленко

1. Назначение и общая структура системы

При проектировании бортового радио- и электрооборудования (БРЭО), сложных технических объектов разработка и выпуск электрических схем соединения и смещённых схем являются одной из самых трудоёмких работ. В процессе их проектирования необходимо обеспечить электрическое соединение десятков и сотен блоков, каждый из которых может иметь несколько разъёмов, в свою очередь, разъём — это десятки контактов, провода, а кроме того, экраны, плетёнки, жгуты и т.д.

По мере проработки того или иного проекта необходимо выполнять значительное количество модификаций. Процесс этот весьма трудоёмкий и не исключает возможности принятия ошибочных решений, что в значительной степени замедляет доводку изделия.

Система ЭЛМОН, разработанная в рамках САПР БРЭО сложных технических объектов, позволяет автоматизировать этот процесс. Ряд требований на систему изложен в [1]. Система состоит из командной процедуры, связывающей пользователей с операционной системой, управляющего монитора, обеспечивающего общение пользователей с ЭВМ и поддерживающего семантику электросхем, подсистем построения графической и текстовой документации, нормативно-справочных данных и базы данных, в которой хранится необходимая информация.

В случае необходимости можно создавать несколько экземпляров баз данных для использования их в различных проектах.

Пользователи системы не имеют доступа к командному языку ЭВМ, что повышает надёжность функционирования и не требует от пользователей специальной подготовки.

Настройка системы на технологию, принятую в эксплуатирующей организации, коррекция нормативно-справочных данных, регистрация пользователей, использование утилит и ряд других функций выполняются администратором системы.

Обучение пользователей, консультации, подготовка предложений по развитию системы, а также участие в выпуске наиболее важной документации возлагается на группу поддержки системы.

2. Логическая схема данных

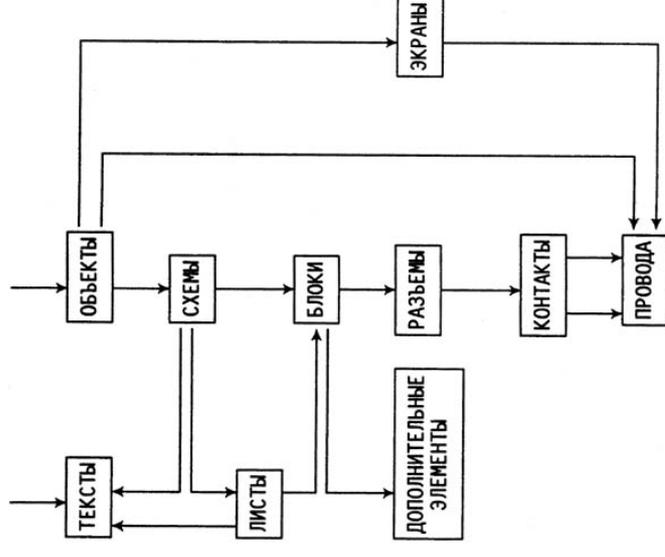
В системе ЭЛМОН принята сетевая организация данных, поддерживаемая СУБД ГЕНБД-4 [2]. Логическая схема (см. рисунок) реализует задачи, сформулированные в разд. 3.

* * *

Основные типы записей (в скобках указана часть содержательных полей по каждому типу):

- объекты (номер объекта, номер модификации, имя объекта);
- схемы (номер схемы, группа оборудования, наименование схемы, рекомендуемое сечение проводов в схеме);
- блоки (тип блока, наименование и обозначение блока, место установки, позиционное обозначение);
- разъёмы (тип разъёма, позиция разъёма);
- контакты (номер контакта, наименование сигнала или цепи);
- провода (бирка провода, марка провода, сечение, длина);
- экраны (тип экрана, номера контактов для заземления);

листы (номер листа, текстовая спецификация), тексты (строка текста).



Логическая схема базы данных.

Для этих типов записей определён ряд наборов типа "владелец набора — член набора". Так, например, каждый объект может включать в себя одну или несколько схем, каждая схема — один или несколько блоков и т.д. Связь между контактами и проводами поддерживается двумя наборами, что позволяет ввести понятие "входящих" и "выходящих" проводов. Это удобно для реализации целого ряда задач.

Для большинства наборов определены ключевые поля (простые и составные) для записей, являющихся членами наборов. При этом блокируется

возможность создания записей-дубликатов в БД. Так, например, в объекте могут создаваться схемы только с отличающимися друг от друга номерами; позиционные обозначения блоков уникальны в схеме-владельце (но могут совпадать в разных схемах) и т. д.

3. Предметная область

В системе ЭЛМОН ставится задача реализовать следующие основные операции над данными.

ОБЪЕКТЫ

создание, модификация объектов (изделий);
уничтожение объекта и всех относящихся к нему данных;
нахождение провода по бирке с восстановлением контекста;
возможность переписи данных по группе схем из одного объекта в другой (в том числе и из одной БД в другую);
контроль корректности данных по объекту.

СХЕМЫ

создание, модификация схем в составе объектов;
уничтожение схемы и всех относящихся к ней данных;
возможность постановки схемы на контроль изменений с выпуском в случае необходимости предварительных извещений;
поддержка понятий "своих" и "чужих" проводов для данной схемы;
задание и коррекция межблочных расстановок для простых жгутов;
выпуск текстовой документации по схемам: таблиц соединений, материальных спецификаций, распаячных листов, вспомогательных документов;
выпуск графической документации: схем соединений и совмещённых схем в виде одного или нескольких чертёжных листов, графических приложений к предварительным извещениям.

ЭЛЕМЕНТЫ СХЕМ (БЛОКИ И СОЕДИНИТЕЛИ)

создание, модификация элементов в составе схемы;
уничтожение элемента и всех относящихся к нему данных;
возможность создания подэлементов в составе элемента (например, соединителей в составе блока);

поддержка различных типов элементов:

блоков — покупных комплектующих изделий;
блоков-коробок (сборочных единиц);
технологических и герморазъёмов;
клеммколодок различных типов;
муфт сращивания;
гермоводов;
минусовых шин и болтов;
контрольных разъёмов;
ШРАПов;
аппаратов защиты;
соединителей различного типа (в том числе с нечисловым обозначением контактов);
элементов принципиальных схем с графическим изображением (реле, выключатели и т.д.);
разъёмов типа "окно" и "клемма";
высокочастотных соединителей и переходов;
сборка фидера, защищаемого аппаратом защиты, проверка типоминала автомата защиты, поверочный расчёт фидера на падение напряжения и токи короткого замыкания, выдача требуемых сечений проводов;

использование нормативных данных по характеристикам элементов различного типа;

привязка элемента к месту размещения на борту ЛА.

КОНТАКТЫ (КЛЕММЫ)

автоматическое создание необходимого количества контактов (клемм) и их нумерация при создании соответствующего элемента (подэлемента);

создание, хранение, модификация наименования сигнала;

возможность нечисловой нумерации контактов (например, для разъёмов типа РПК и ШРАП).

ПРОВОДА

создание, хранение, модификация проводов с учётом используемых нормативов (марка провода, сечение);

присоединение проводов к контактам с проверкой соответствия диаметра контакта сечению провода;

выработка бирки провода по технологии, принятой в данной организации;

учёт особенностей присоединения:

перемычек;

радиокабелей;

минусовых проводов (в том числе при заземлении на корпус);

на клеммоколодки, муфты сращивания;

на технологические и герморазъёмы;

от распределительных устройств;

под болтовые соединения;

возможность переадресации проводов;

отсоединение проводов или перевод в резерв (в схемах, поставленных на контроль изменений);

автоматическое определение типов наконечников и заделки;

возможность выбора ответного адреса при присоединении провода различными способами (табличным, позиционным и т.д.);

возможность указания на свивку проводов (в том числе с минусовым проводом), на использование экранирующих плетёнок;

задание способа заземления проводов и плетёнок.

ЛИСТЫ

создание, хранение, модификация чертёжных листов;

задание поблочного состава листа;

задание текстовой спецификации к листу;

автоматическое определение габаритов блоков и разъёмов и их размещение на листе;

автоматическое построение перечня элементов, штампов;

возможность изменения пользователем привязки блоков к листу, их ориентации;

обеспечение разделения листа на фрагменты в связи с техническими возможностями графопостроителей.

4. Монитор ввода/модификации данных

Управляющий монитор обеспечивает ввод пользователями исходных данных, контроль их корректности, поддержание семантики электросхем, связь с БД.

Все действия пользователя, в том числе и выход на подсистемы построения конструкторских документов, осуществляются через выбор одной из команд из меню. Меню разбиты на ряд уровней (объектов, схем, блоков, разъёмов, контактов, листов, аппаратов защиты) в соответствии с логикой работы пользователя. Монитор в большинстве случаев предлагает пользователю тот или иной вариант действий или значения данным по умолчанию. В связи с тем, что в различных организациях технология проектирования и принятая

семантика существенно отличаются друг от друга, в системе ЭЛМОП используются настроенные файлы, доступные администратору системы. Ряд режимов доступны пользователю во время текущего сеанса работы. Ввод данных в диалоговом режиме обеспечивается с помощью пакета программ ввода, основанного на UMG-технологии. При вводе данных (пока не нажата клавиша RETURN) можно редактировать уже введенные символы. С целью протоколирования действий пользователя предусмотрена возможность одновременно с вводом данных записывать их в специальный журнальный файл. В режиме тестирования вместо непосредственного ввода данных можно использовать уже имеющиеся журнальные файлы.

Основные программы ввода:

- ввод текстовых данных;
- ввод целого или вещественного числа;
- ввод целого числа с контролем (целое число проверяется на принадлежность к диапазону IMIM...IMAX, указываемому при обращении к программе);
- ввод вещественного числа с контролем.

5. Графическая подсистема. Построение графической КД

В каждой электросхеме можно определить один или несколько чертёжных листов, в каждый из которых входит один или несколько блоков. Чертёжный лист представляет собой прямоугольник, ширина которого фиксируется, а длина определяется после размещения всех элементов. Лист разбивается на форматки, то есть прямоугольнички размером $21,0 \times 29,7$ см. Форматки размещаются в 1 или 2 ряда. Ряды обозначаются буквой, а номер форматки в ряду — цифрой. Таким образом, каждая точка листа имеет графический адрес, который определяется рядом и номером форматки. Графический адрес элемента определяется его левым верхним углом. На листе могут располагаться блоки

разных типов, тексты, перечень элементов, штампы.

Алгоритм размещения включает в себя:

- автоматическое определение габаритов блока;
- выбор свободной точки листа. (Для первого из размещаемых блоков свободная точка — левый верхний угол листа. Левый верхний угол блока размещается в свободной точке. Если блок не помещается на листе, то свободной точкой считается верхняя левая не занятая другими блоками точка листа. В противном случае свободная точка располагается ниже размещённого блока);
- размещение в правой части листа текстовой спецификации, перечня элементов (для 1-го листа схемы) и штампов. Перечень элементов и штампы строятся автоматически

Блок—ПКИ (покупное комплектующее изделие) изображается прямоугольником, содержащим наименование и позицию блока. Габариты определяются размерами входящих в этот блок разъемов. Алгоритм размещения разъемов в блоке аналогичен алгоритму размещения блоков на листе, за исключением того, что разъемы могут быть развернуты на 90 градусов. Разъем представляется в табличном виде. В каждой графе таблицы содержится информация о контакте или проводе. Строка таблицы соответствует контакту и присоединённому к нему проводу. Пользователь имеет возможность отказаться от некоторых граф. Над таблицей указывается тип, позиция разъёма, условно изображается тип соединителя ("вилка" или "розетка"). Возле каждой строки показывается провод и экран. Провода могут входить в экранирующие плёнки, могут быть свиты. Изображения свивок, экранов не должны пересекаться, поэтому при необходимости они располагаются на разных расстояниях от таблицы. Комбинаторная задача размещения экранов решается эвристической процедурой.

Блок-коробка (сборочное изделие) отличается от ПКИ тем, что может включать элементы принципиальных схем. Они размещаются аналогично разбёмам. Габариты и способ их изображения задаются во внешнем файле и могут быть дополнены администратором системы.

Другие элементы в значительной степени получают комбинациями вышеописанных.

В системе имеется специальный настроечный файл, позволяющий варьировать способы изображения элементов и листов в целом. Например, нумерация форматок слева направо или справа налево, изображение или пропуск кон тактов, не содержащих провода, и т.д. Одним из параметров является максимальная длина графопостроителя. Лист автоматически разбивается на фрагменты с длиной меньше, чем заданная. При этом место разбиения выбирается между блоками (разбёмами), чтобы не задевать элементы чертежа.

После того как система автоматически сформирует чертёжный лист, пользователю предоставляется возможность внести некоторые изменения. С этой целью на экране дисплея условно изображается лист и его элементы. Если длина листа велика, то изображается его часть, причём пользователь может выбрать интересующий фрагмент. Имеются следующие операции:

- перенос блока (разбёма). Курсором выбирается элемент — по близости левого верхнего угла, затем указывается его новое место размещения;
- выделение (уничтожение) свободного места на чертеже либо внутри блока. Место указывается двумя точками. Если точки находятся внутри блока, то изменяются его габариты, а если нет, то изменяется длина листа;
- поворот разбёма, дополнительного элемента. Выбранный элемент поворачивается на 90 градусов;
- установка (отмена) границ фрагментов чертежа;

выбор окна для увеличенного изображения на экране; измерение расстояния между двумя точками чертежа.

Графическое отображение чертежа представляется в системе в виде последовательного файла, каждая строка которого включает название графической функции и её аргументы. Исключение — текст, занимающий две строки.

Например:

M 0.0 1.1

D 1.1 1.1

T 1.1 1.1 0.5 0.0 5

ПРОБА

Первая команда означает, что перо нужно поместить в точку с координатами (0.0, 1.1). Вторая команда — провести прямую из текущей точки в точку (1.1, 1.1). Третья — написать текст 'ПРОБА' из 5 букв высотой 0.5 см с наклоном 0.0 градусов из точки (1.1, 1.1). Такое представление позволяет, во-первых, не зависеть от используемой системы программирования, во-вторых, пользователь может редактировать свой чертёж.

6. Построение текстовой КД

Система ЭЛМОН позволяет автоматически выпускать таблицы соединений (с разбивкой всего множества проводов схемы на простые жгуты; подсчётом суммарных весов жгутов; определением типа заделки и наконечников), материальные спецификации (суммирующие используемые в схеме стандартные изделия и детали, а также расход материалов), распаячные листы, предельные извещения (с графическими приложениями), вспомогательную документацию. Поддерживается возможность настройки КД на различные формы.

7. Нормативно-справочные данные

Для обеспечения возможности оперативного изменения данных нормативно-справочного характера в системе ЭЛМОН разработаны специальные программные средства. Возможные изменения в составе этих данных производятся администратором системы и не требуют никаких модификаций системы. Порядок расположения строк информации в каталогах произволен. Администратор системы может вводить новые строки, исключать неиспользуемые.

Каталог соединителей (разъёмов) включает в себя данные о типах разъёмов, используемых в данной организации, их массе и диаметре всех контактов. Каталог состоит из строк вида:

<тип соединителя> <масса в кг> <структура соединителя>

Например,

'СНЦ23-45/39Р-13-В' 0.104 '1с2b38a2c2a'

Запись 1с2b38a2c2a означает:

- 1 контакт диаметром 2.0 мм (контакт № 1)
- 2 контакта диаметром 1.5 мм (контакты № 2 и № 3)
- 38 контактов диаметром 1.0 мм (контакты № 4 ... № 41)
- 2 контакта диаметром 2.0 мм (контакты № 42 и № 43)
- 2 контакта диаметром 1.0 мм (контакты № 44 и № 45)

Каталог высокочастотных соединителей содержит сведения о типах высококочастотных соединителей (ВЧ), виде соединителя (вилка/розетка), а также о типах рекомендуемых радиокабелей для подсоединения к данному типу соединителя.

Каталог проводов хранит данные о марках проводов, их характеристиках и типах применяемых наконечников. В первой содержательной строке указываются допустимые сечения проводов [мм кв.]:

1 16 0.2 0.35 0.5 0.75 ... 70.0 95.0

Здесь символ "1" — служебный, "16" — общее количество сечений, далее — их значения. Затем следуют группы данных по каждой марке провода, состоящие из двух частей: общих для данной марки сведений и ряда строк с характеристиками по каждому применяемому сечению. Общие сведения состоят из одной строки, в которой указаны: марка провода, технические условия, температурный диапазон применения данной марки провода. Пример:

1 'БИФН' 'ТУ 16-505.620-76' -60.0 200.0

По каждому сечению приведены по две строки данных. В них указываются:

сечение провода [мм кв.];

наружный диаметр провода [мм];

вес 1-го метра провода [г/м];

омическое и реактивное сопротивление провода [ом/м];

тип наконечника для заделки провода на клеммколodки, минусовые панели и при присоединении под болт и их веса.

Данные по радиокабелям размещаются в конце каталога проводов.

Каталог экранирующих плетёнок включает в себя данные по типоразмерам и весам экранирующих плетёнок. Каждый типоразмер рассчитан на некоторый диапазон значений эквивалентного диаметра жгута, определяемого по формуле:

$$D = 1.3 * [2.0 / 3.1415 * D_{sum} * (D_{min} + D_{max})]^{**0.5}, \text{ где}$$

D_{min} – минимальный диаметр провода жгута;

D_{max} – максимальный диаметр провода жгута;

D_{sum} – сумма диаметров всех проводов жгута.

Каталог клеммколodок и минусовых панелей содержит в себе три группы данных с различным форматом: для клеммколodок, минусовых шин и минусовых болтов. Каждая строка данных по клеммколodкам имеет вид:

<служебное слово "КК"> обозначение клеммколодки>
<примечание> <количество клемм> <наименование>.

Примеры:

КК КОЛОДКА 1 ОСТ 113341-78 ' ' 2 'КЛЕММНАЯ КОЛОДКА'

КК *-ОСТ 112273-77 'МУФТА СРАЩИВАНИЯ' 1

_-*ОСТ 103868-77

Строки данных по минусовым шинам имеют вид:

служебное слово "MSH"> обозначение МШ> <кол-во клемм>

Например,

MSH '30.7200.1815.901' 4

Строки данных по минусовым болтам имеют вид:

<служебное слово "MB"> <обозначение МБ> <сеч.проводов>

Например,

MB '1-2-ОСТ1 14092-81' '2.5; 4.0'

Каталог муфт сращивания содержит перечень типоразмеров муфт и трубок для различных вариантов присоединения проводов. Помимо сечений проводов, в каждой строке данных указывается:

тип МС; типоразмер МС; номер МС; вес МС; тип и вес трубки.

Например,

1.5 0.0 4.0 0.0 1 4 5 1.1 2 0.95

При этом МС получает обозначение 5-ОСГ112273-77, трубка — 2-ОСТ1 14404-85, заделка МС — 1-4-* ОСП 03868-77 (вместо символа "*" подставляется цифра "1" или "2" в зависимости от марки используемых проводов).

8. Общие характеристики

Система ЭЛМОН реализована для ЭВМ типа СМ-1700 на языке Фортран-77. Объём — около 20 тыс. строк.

среднего класса, построенные с использованием системы ЭЛМОН, составили 12 тыс. форматок А11, что сэкономило 14400 нормочасов.

Литература

1. Aviation Week, 5/Ш 1973.
2. Прибытков Ю.Д., Шиленко В.И. Программный комплекс ЭЛМОН. Наст. сб.

СОДЕРЖАНИЕ

	Стр.
Иерархия задач проектирования. – П.С. Краснощёков, Ю.А. Флёров	3
Программный комплекс формирования облика летательных аппаратов. – Л.Л. Вышинский, О.С. Самойлович, Ю.А. Флёров	24
Структура моделей в задачах проектирования. – Л.Л. Вышинский	43
Инструментальные средства САПР – Л.Л. Вышинский, И.Л. Гринёв, В.И. Шиленко, Н.И. Широков	52
Средства управления данными в САПР – И.Л. Гринёв, Н.И. Широков	71
Основы построения программного комплекса лётно-технических характеристик в САПР ЛА. – В.П. Катунин	81
Весовой анализ и контроль в САПР ЛА. – С.И. Скобелев, Н.И. Широков	92
Автоматизация проектирования электросхем – М.Ю. Шапиро, В.И. Шиленко	101
Программный комплекс ЭЛМОН – Ю.Д. Прибытков, В.И. Шиленко	106

Задачи и методы автоматизированного проектирования в авиастроении

Редактор Л.А. Лебедева. Техн. редактор П.А. Швайко
Корректурa авторов

Подписано в печать 11.11.90. Заказ 30

Тираж 301 экз. Формат бумаги 60×90¹/₁₆

Уч.-изд. л. 5,44. Усл.-печ. л. 7,56. Цена 1 р. 10 коп.

Отпечатано на ротационных в ВЦ АН СССР
117333, Москва, ул. Вавилова, 40