

МОДЕЛИ КОГНИТИВНЫХ ПРОЦЕССОВ (Вычислительные системы)

1997 год

Выпуск 158

УДК 519.68

ОБ ОДНОМ СПОСОБЕ ОРГАНИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ТЕХНОЛОГИЧЕСКОМ КОМПЛЕКСЕ РЕШЕНИЯ ЛОГИЧЕСКИХ ЗАДАЧ ¹

А.А.Москвитин

Процесс решения задач в описываемом технологическом комплексе проходит несколько стадий исполнения: описание задачи в терминах предметной области, спецификация задачи на формальном языке, выбор исполнительного алгоритма, создание программных средств и собственно решение задачи. Среди этих этапов выделим один — создание программных средств системы по ее формальному описанию на языке спецификаций задачи, извлеченному в процессе диалога с пользователем.

Для этих целей разработан специальный язык, состоящий из набора исходных императивов и набора исходных предикатов. Указанные наборы реализованы в виде специальных базисных модулей или **В-модулей** и хранятся в **Базе Данных и Программ (БДП)** технологического комплекса.

Задача, сформулированная пользователем средствами языка спецификации задач, преобразуется в программную систему методом извлечения из доказательств и конструируется из **В-модулей**.

Процесс извлечения исходной постановки задачи пользователя, ее формализация, конструирование программ и

¹Работа поддержана РФФИ, грант № 96-06-80970.

их исполнение осуществляется в технологическом комплексе решения задач (не путать с технологией программирования), состоящем из следующих подсистем:

- подсистемы осуществления диалога с пользователем по извлечению из него спецификации задачи;
- подсистемы синтеза исполнительской среды программной системы из В-модулей по полученной спецификации;
- подсистемы исполнения и анализа результатов решаемой задачи.

Проект подсистемы и опытный образец диалога с пользователем по формированию спецификации задачи описаны в работе [1]. Теоретическое описание языка спецификаций задач данного технологического комплекса решения задач на компьютере приводится в работе [2]. В данной работе будет детально рассмотрен вопрос организации программного обеспечения **базисного уровня технологического комплекса решения задач (ТеКоРЗ)**.

1. Базисный уровень технологического комплекса решения задач

Техническую основу базисного уровня составляет база данных и программ, а также подсистема управления ею. Поскольку особенностью данного комплекса является модульность схем исполнения, то естественно, что БДП технологического комплекса состоит из некоторой иерархии исполняемых В-модулей.

В-модуль — это описание того набора исходных операций и отношений в терминах которых будет исполняться данная программа.

В-модуль в ТеКоРЗ не совсем обычен. Он может работать в трех различных режимах: обычном вычислительном, проверочном и генерирующем. Для этого в В-модуль вводятся три отдельных типа данных: обрабатываемые данные; данные, задающие условие исполнимости модуля, и результирующие данные.

В **обычном режиме** на вход В-модуля подаются обрабатываемые данные и данные условия выполнимости

(которые определяет пользователь). На выходе имеем некоторые результирующие данные.

При проверочном режиме на вход В-модуля подаются обрабатываемые данные и предполагаемые результаты. На выходе имеем условие проверки исполнимости модуля.

Генерирующий режим предполагает задание на входе В-модуля условия выполнимости и предполагаемых результатов его выполнения. На выходе имеем все возможные обрабатываемые данные, удовлетворяющие заданным условиям на входе В-модуля.

Естественно, что развитость любой системы программирования определяется наработанным программным продуктом, которым пользователь может распоряжаться как набором исходных примитивов. Для ТеКоРЗ это означает многообразие В-модулей.

Особое положение среди них занимают модули, которые носят инструментально-технологический характер. Такие модули могут быть использованы не только для создания программных систем, решающих задачи проблемной области, но и служат средством расширения, модификации самого языка спецификации задач.

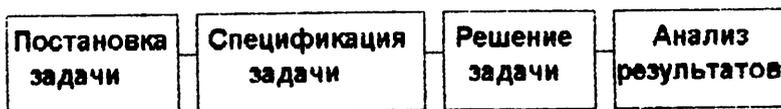
Задача в ТеКоРЗ — это специального вида логико-математическое описание, допускающее эффективное "исполнение". Основными конструкциями языка описания задач являются модули и запросы. Модули могут представлять собой конструкции, которые помимо определенных могут содержать дополнительную информацию. Эта дополнительная информация предназначена либо человеку и носит, таким образом, поясняющий характер; либо вычислителю и, следовательно, в определенном смысле является "подсказкой", предназначенной для улучшения хода вычислений или машинного доказательства.

Описание задачи в ТеКоРЗ [2] задается как пара (T, Res) , где T — формальное описание предметной области, а Res — интеллектуальный ресурс пользователя, рассуждающего в рамках данной аксиоматической системы, представим тройкой $Res = (m_1, m_2, m_3)$, где m_1 — наи-

большая длина доказательств в данной системе, все еще имеющих достаточно высокую (заранее фиксированную) степень убедительности для данного человека, m_2 — наибольшая длина последовательностей символов алфавита языка данной системы, все еще имеющих достаточно высокую (заранее фиксированную) степень безошибочного распознавания данным человеком как формулы (или не формулы) языка данной системы; m_3 — то же, что и m_2 , но применительно к термам.

По типам решаемых задач и видам описания модули в ТеКоРЗ можно разбить на следующие группы: В-модули, Т-модули, I-модуль. В-модуль — основной базисный модуль технологического комплекса решения задач. Из В-модулей, как из "кирпичиков", конструируется программная система, призванная решить задачу пользователя; S-модуль — спецификационный модуль, описывающий задачу на формальном уровне; Т-модуль — модуль описания теории; I-модуль — инструментальный модуль ТеКоРЗ, выполняющий вспомогательные функции взаимодействия пользователя с компьютером.

Схема решения задач в общем виде может выглядеть следующим образом:



1.1. **В-модули.** Вначале несколько слов о структуре В-модуля. Исходный текст В-модуля в ТеКоРЗ имеет четко выраженные две части: описательную и исполнительную.

В описательной части представлены все особенности взаимодействия данного В-модуля с другими В-модулями в изготавливаемой программной системе, решающей задачу пользователя.

В исполнительной части задается последовательность вычислений в данном В-модуле, зависящая от особенно-

стей его использования в программной системе. Таких особенностей использования может быть три: **прямое вычисление** — по исходным данным и априори заданным особенностям вычислительного процесса (условиям счета) получаем результат решения задачи; **проверочное вычисление** — по входным данным и известным результатам вычислений проверяются условия применения данного В-модуля; **генерация входной последовательности** — по известным результатам счета и априори полученным сведениям об использовании В-модуля (условиям его исполнения) синтезируются все возможные входные последовательности данных, приводящие к требуемым результатам вычислений.

Согласно перечисленным моделям вычислений, каждый В-модуль (в отличие от обычных модулей) имеет три точки входа-выхода (см. рис.1).

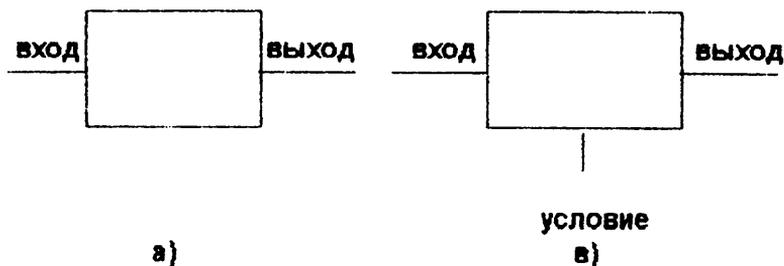


Рис.1.

Формальное описание В-модуля в ТеКоРЗ отличается от общепринятого и выглядит следующим образом. В описательной части В-модуля отображается вся информация, отвечающая на вопрос **ЧТО** делает данный В-модуль (рис.2), каковы его основные функции, с какими структурами данных он работает и т.п. В исполнительной части В-модуля сосредоточен весь перечень исполнительных команд, указан порядок их следования, позволяющий ответить на вопрос **КАК** должен протекать процесс

вычислений в данном модуле в зависимости от одной из трех заданных схем его вычислений.

Таким образом, структура В-модуля может быть представлена следующим описанием:

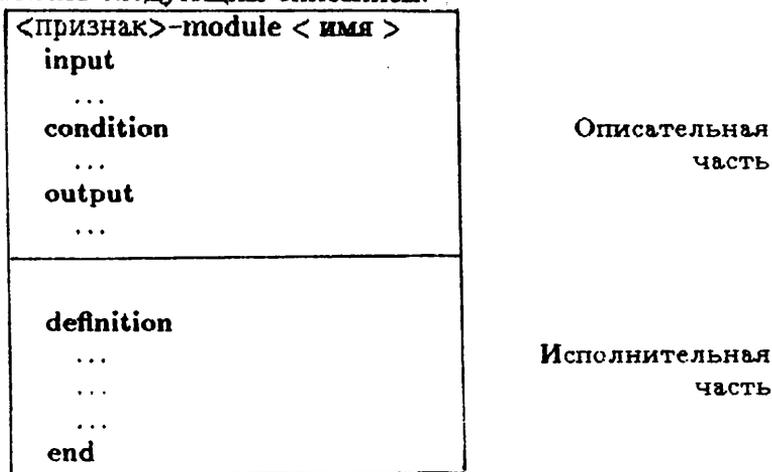


Рис. 2.

На рис.2:

<признак> указывает на тип модуля (базисный, спецификационный, теоретический, инструментальный и т.п.);

<имя> — собственное имя модуля;

input — входные параметры;

condition — условие применения;

output — выходные параметры;

definition — ключ-разделитель описательной и исполнительной частей модуля;

end — признак конца исполнительной части модуля.

Исполнительная часть В-модуля устроена также не в традиционном стиле. Она складывается как-бы из "кирпичиков" — фрагментов модуля, выполняющих логически завершенную часть вычислений. Так, например, в модуле вычисляющем корни полного квадратного уравнения, таким "кирпичиком" может быть совокупность команд вычисления дискриминанта. Этот прин-

цип (конструирования из "кирпичиков") распространяется и на процесс конструирования всей программной системы. Элементарный "кирпичик" В-модуля будем именовать ELS_i . Такой способ организации модулей позволяет динамически конструировать, требуемое по условиям задачи, программное обеспечение оптимальной архитектуры.

1.2. **Управляющий массив.** В процессе конструирования программной системы в соответствии со спецификацией задачи, сделанной пользователем, происходит трансформация указанных В-модулей в программную систему по следующим правилам. Все описательные части модулей извлекаются из В-модулей и объединяются в одном месте, называемом **управляющим массивом**. Из исполнительных частей строится схема вычислений, согласно заданной схеме вычислений для решения задач пользователя (см.рис.3).

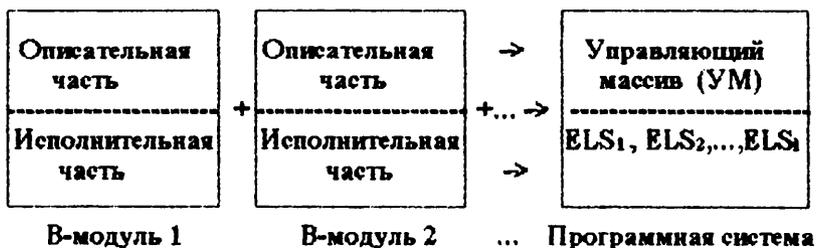


Рис.3.

2. Базисный уровень организации программного обеспечения

Как видно из названия, настоящий раздел будет посвящен более подробному описанию тех средств языка спецификации задач, которые ориентированы на программирование в терминах В-модулей и решение задач в ТеКоРЗ.

Эту часть языка можно рассматривать как самостоятельный язык (тоже логического типа). Другая точка зрения на данный фрагмент языка — это язык **манипулирования данными** некоторой системы управления базами данных следующего смысла. На всю совокупность базовых средств (процедуры и соответствующие им описания опций, их логические свойства, имена функций и предикатов, их связи с другими процедурами, стандартные В-модули и т.п.) можно смотреть как на некоторую **сигнатурную базу данных**. Тогда описание В-модуля представляет собой запрос к сигнатурной базе данных, в результате обработки которого должен быть сгенерирован нужный В-модуль (или связанная совокупность В-модулей). Именно под таким углом и будет рассматриваться базисный уровень языка спецификации задач в следующем подразделе.

Решение задач на базисном уровне. Задача в ТеКоРЗ, как уже упоминалось выше, представляет собой пару (T, Res) . Эта пара в свою очередь распадается на следующее:

- 1) *совокупность модулей M_1, M_2, \dots, M_k* (вместе с описанием их взаимосвязей) в виде некоторой схемы исполнения;
- 2) *запрос;*
- 3) *описание контекста задачи, представляющее собой набор значений для входных переменных запроса; (с каждым запросом может быть связано несколько контекстов, а описание каждого конкретного контекста назовем пунктом).*

Стандартные В-модули в блоке могут не описываться, но их имена и имена их структурных конструкций могут использоваться как в модулях, так и в запросе.

В соответствии с такой структурой задачи на базисном уровне (**базисной задачи или И-задачи**) процесс ее решения с реализационной точки зрения естественно разделить на три этапа:

- а) *конструирование В-модулей в соответствии с их спецификацией;*

- б) конструирование схемы решения задачи;
- в) решение задачи в соответствии с запросом и очередным пунктом.

Реализация первых двух из указанных пунктов предполагает работу с распределенной базой данных (в широком смысле слова), в которой должно содержаться следующее:

- 1) "элементарные" программные конструкции (ELS) В-модулей (своеобразный аналог термов в программировании), соответствующие содержанию базисных сигнатурных символов;
- 2) стандартные В-модули, реализующие типовые схемы решения задач проблемной области (основные алгоритмы различных разделов информатики: работа с числами, обработка символьных данных, графика и т.п.);
- 3) специфические В-модули класса задач пользователя;
- 4) вспомогательные инструментальные и служебные программные средства.

Для работы с такой базой данных необходимо иметь языки для описания данных и манипулирования с ними. В ТеКоРЗ предусмотрен двухуровневый язык манипулирования данными (ЯМД-1 и ЯМД-2) и язык описания данных. Указанные языки базируются на основных принципах языка спецификации задач [2] и являются его специальными диалектами. Синтаксис языка описания данных и языка манипулирования данными содержит основные синтаксические категории языка спецификации задач.

Схема работы с такой распределенной базой данных приведена на рис.4.

3. Технологический этап решения задач в ТеКоРЗ

Технологический этап решения задач в ТеКоРЗ представляет собой этап, на котором разрабатываемый метод решения Т-задач приобретает новые черты конкретного управляемого процесса организации человеко-машинной

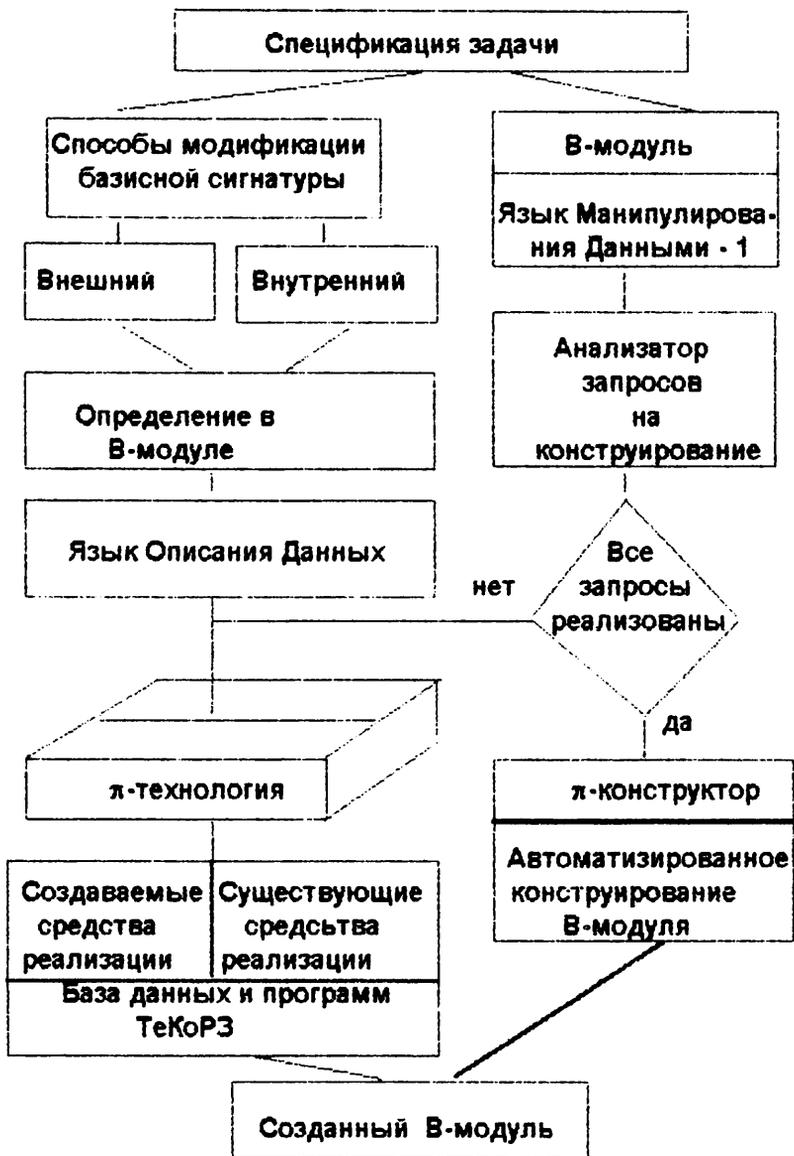


Рис. 4. Схема конструирования V-модуля в ТеКоРЗ.

системы (Т-системы), позволяющей последней эффективно решать Т-задачи, рассматриваемого класса. Таким образом, решение Т-задачи — это технологически подерживаемый акт деятельности Т-системы, а метод решения Т-задачи представляет собой технологический процесс, который может быть по разному реализован в различных Т-системах. Заметим, что из вышесказанного вытекает, что не существует универсальных технологических процессов, так как принятая система семантических представлений может оказаться совершенно неадекватной для другого класса Т-задач.

Отличительной особенностью Т-задач является "неизбежная" реальность их решений. Другими словами, реализации программы этих задач должны быть не просто принципиально осуществимыми или конструктивными, а должны удовлетворять еще и определенным и зачастую весьма жестким требованиям. Часть этих условий носит достаточно четкий количественный характер. Таковыми, например, являются рекурсивные ограничения на время счета (скажем, среднее или асимптотическое) и/или требуемый объем памяти, т.е. ограничения на временную и емкостную сложность программы. Другие требования носят совершенно новый характер — качественный. К этим требованиям относятся, например, требования на мобильность, точность, завершенность программы и т.п. Иногда эти требования несущественны и мы выявляем их наличие, когда начинаем говорить о достоинствах созданной программы. Но иногда заказчик с самого начала требует не просто решение некоторой Т-задачи, а "решение, которое должно быть надежным". В этой ситуации программирование выступает как метод некоторой "квазиоптимизационной" задачи. Для того, чтобы решить такую задачу, необходимо вначале уточнить представление заказчика (да и свое собственное) о том, какой смысл он вкладывает в подобные требования. Возникает потребность в методике формирования исходных требований на программу, которые можно рассматривать как часть спе-

цификации Т-задачи. Выявленные требования имеют серьезное значение в выборе дальнейших как проектных, так и кодировочных решений. При этом чрезвычайно полезным является опыт решения похожих Т-задач. Этот опыт, а также некоторые теоретические рассуждения, могут нам подсказать, что некоторые требования трудно осуществимы (требуют больших трудозатрат), другие сами по себе реализовать легко, но в сочетании с другими они неосуществимы и т.д. и т.п. Желательно при этом знать — какое значение придает этим исходным требованиям заказчик, насколько он может отказаться от них. Так возникает следующая содержательная модель методики формирования исходных требований (см. рис. 5).

С каждым требованием (предполагается, что их набор достаточно универсален; появление новых требует соответствующей модификации проектировочной базы знаний) связывается совокупность вопросов, уточняющих содержание этой характеристики. Одним из таких вопросов является вопрос, уточняющий важность для заказчика удовлетворения этого требования в целом, либо каждой части вопроса, связанного с требованием. Каждому вопросу ставится в соответствие набор ответов (этот набор либо заранее фиксирован, либо открыт). За ответами стоят конструкторские решения, которые могут осуществляться либо на этапе проектирования, либо (чаще всего) на этапе кодирования. Из предыдущего опыта или из чисто теоретических соображений может следовать, что некоторые наборы ответов несовместны, другие совместны, но требуют больших трудозатрат и т.д. Предполагается, что такая информация также имеется в базе знаний технологической системы Syst(K). Тогда пользователь системы должен быть информирован об этом и либо принять решение об изменении своих ответов (вплоть до отказа от требования), либо предоставить это сделать системе, которая по имеющимся у нее предварительным оценкам желательности свойства и имеющимся знаниям модифицирует данный набор ответов и информирует об этом пользователя.

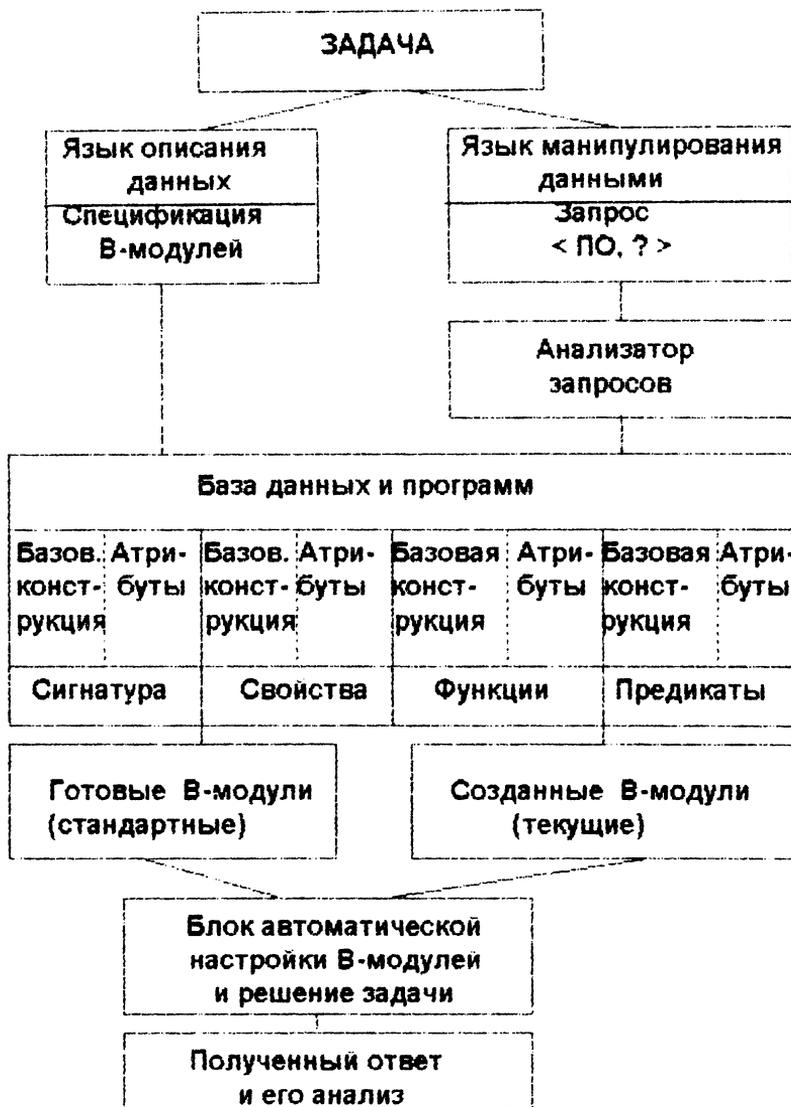


Рис. 5. Схема решения задач в ТеCoP3.

Подобная модель допускает точную математическую постановку и алгоритмическое решение. Далее мы перейдем к изложению того, как мыслится содержательная процедура уточнения исходных требований на программную систему.

Работа по определению исходных требований осуществляется на основе определения спецификации задачи пользователем и согласования ее с заложенными в набор стандартными (допустимыми) проектными решениями (базы знаний Syst(K) системы TeKoPЗ).

Результатом рассматриваемого этапа проектирования являются файлы проектировочных решений формальной постановки пользователем T-задачи, готовые к конструированию из них программной системы. Эти файлы доступны пользователю для уточнения и модификации. Их исполнение (или сообщение пользователю о невозможности исполнения) начинается с момента выдачи пользователем запроса на решение сформулированной им задачи.

Среди всех уточнений задачи пользователя существует ряд ограничений на успех осуществления проекта решения задачи, связанных с техническими возможностями использования предполагаемых типов компьютеров и удобства процесса решения задачи. Приведем основные из них, влияющие на качественное решение T-задачи.

Свойства качественных решений T-задач. При оценке качества решения некоторой T-задачи требуется ответить, по крайней мере, на следующие три вопроса.

1. Возможно ли использовать полученное решение в исходном виде?
2. Чем определяется удобство эксплуатации программной системы?
3. Возможно ли применить созданную программную систему в иных условиях?

При ответе на первый вопрос необходимо:

- понять, что способна делать программа;
- получить достоверные результаты, выявить и устранить источники недостоверности и ненадежности;

- обеспечить рациональное и эффективное использование человеческих и машинных ресурсов.

Для ответа на второй вопрос необходимо:

- понять структуру программы;
- разработать и реализовать требуемые модификации;
- провести тестирование для обеспечения гарантированной надежной работы;
- обеспечить рациональное и эффективное использование человеческих и машинных ресурсов.

Отвечая на третий вопрос, необходимо:

- оценить исходную полезность программы;
- оценить степень удобства эксплуатации;
- оценить ее мобильность;
- приспособить работу программы к иным условиям.

Как выделить свойства качественных Т-задач приводится в работе [1].

Л и т е р а т у р а

1. МОСКВИТИН А.А. Среда спецификационной деятельности СИГМА-ТЗ // Теория вычислений и языки спецификации.- Новосибирск, 1995.- Вып.132: Вычислительные системы.- С.76-94.

2. КАЗАКОВ Е.В., МОСКВИТИН А.А., САМОХВАЛОВ К.Ф. Проект разработки языков спецификаций задач, ориентированных на пользователя // Настоящий сборник. - С.63-94.

Поступила в редакцию
19 декабря 1996 года