

Черкасский М., Саченко А.

ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ НА ОСНОВЕ ТЕОРИИ СЛОЖНОСТИ КОМПЬЮТЕРНЫХ АЛГОРИТМОВ

Введение. Теория компьютерных систем, несмотря на её огромное влияние на развитие науки и техники, до сих пор находится в состоянии далёком от завершения. В основе современной технологии проектирования компьютерных средств доминируют эвристические подходы. К ним: относятся архитектурные методы разработки компьютерных средств и методы построения абстрактных алгоритмов. Эти методы сориентированы на оптимизацию технических характеристик компьютерных систем.

Сферой приложения архитектурных методов является разработка технических решений на всех иерархических уровнях вычислительных систем и оптимизация их по производительности и объёму оборудования [1].

Сферой приложения абстрактных алгоритмов является: анализ множества задач, которые подлежат компьютерной реализации, конструирование эффективных алгоритмов, разработка набора команд системы. Исследование абстрактных алгоритмов проводится с целью минимизации временной сложности работы системы [2].

Эти два подхода не замещают, а лишь дополняют друг друга. Сначала методами абстрактных алгоритмов разрабатываются алгоритмические основания проекта. После этого архитектурными методами проектируются аппаратно-программные средства компьютерной системы. Методы абстрактных алгоритмов и архитектурные методы по способам синтеза и критериям оценки характеристик компьютерных систем существенно различаются. Развитие этих методов происходит независимо друг от друга. Всё это не способствует созданию единой теории вычислительных систем, которая бы объединяла идеи построения и функционирования аппаратных и программных средств с идеями конструирования алгоритмов эффективных для решения заданного множества задач.

Проблема разработки теории вычислительных систем может быть решена с использованием принципиально новой модели компьютерных средств, называемой SH-моделью алгоритма (SH – Software/Hardware) [3]. В статье проводится сравнение эффективности разработки компьютерных систем архитектурными методами, методами формальных алгоритмических систем и методами, основанными на использовании SH-модели алгоритма. В результате сравнения подчеркиваются преимущества SH-модели алгоритма для синтеза, анализа и оптимизации современных компьютерных систем.

Понятие “архитектура компьютеров”. В практике исследования и проектирования компьютерных средств на системном и функциональном этапах используются архитектурные методы. Слово “архитектура” в вычислительную технику пришло из строительства, в котором оно означает “искусство строить и украшать здания” [4]. В вычислительную технику термин “архитектура компьютеров” был введен фирмой IBM в конце 50-х годов двадцатого столетия. Сначала он означал возможность реализовывать программы пользователей последовательно разрабатываемых моделей компьютеров некоторого семейства. Причем компьютеры семейства могли различаться величиной производительности, объемом оборудования и значениями других характеристик. Объединял модели семейства набор команд младшей модели. Позже содержание словосочетания “архитектура компьютеров” расширилось. Оно стало означать, кроме набора команд и представление данных также структуры потоков данных, интерфейсы, способы адресации, аппаратное и программное обеспечение и их компоненты, структурные построения на всех иерархических уровнях компьютерной системы. Заметим, что синтез принципиальных схем элементов, а также конструкторская разработка СБИС в архитектурное проектирование не входит. При сравнении

методов архитектурного проектирования с другими технологиями проектирования, в которых модель является основным объектом исследования, для единообразия в терминологии будем пользоваться термином “архитектурная модель”.

Графическим образом архитектурной модели является схема представления компьютерных средств на одном из иерархических уровнях: принципиальная, логическая, функциональная, структурная.

Архитектурная модель. Теоретическая база архитектурных методов проектирования ограничена главным образом методами имитационного моделирования. Например, на системном этапе решаются задачи оптимизации производительности и объема оборудования системы. Здесь используются вероятностные модели систем массового обслуживания. Разработка функциональных схем базируется практически полностью на исследовании имитационных моделей операционных устройств, узлов, процессоров и систем. На адекватность синтезируемых моделей требованиям пользователя в значительной степени влияют творческие способности исследователей, знания известных технических решений.

Процесс разработки технического объекта состоит из трех операций: синтеза, анализа, оптимизации. При синтезе компьютерного объекта руководствуются характеристиками, которым должен удовлетворять объект разработки. Для архитектурного проектирования такими характеристиками являются: производительность компьютерных средств, время срабатывания устройств (временная сложность), объем оборудования и количество команд микропрограммы. Под объемом оборудования понимают количество транзисторов либо вентилях больших интегральных схем, который и называют аппаратной сложностью.

Синтез компьютерных средств выполняется структурным или параметрическим способами. Исходными данными для структурного синтеза является совокупность неформальных алгоритмов, заданных в виде блок-схем, функциональных зависимостей, словесных форм и др. Исходными данными для параметрического синтеза является уже существующая архитектурная модель, характеристики которой нужно улучшить.

Проведение анализа позволяет получить значения характеристик синтезированной модели. Если эти значения в заданных пределах совпадают с требуемыми значениями характеристик, то разработка модели закончена. Если нет, то изменением параметров на этапе оптимизации достигают необходимых значений характеристик. Параметрами архитектурной модели являются технические решения компьютерных средств.

Архитектурная модель на сегодня является основным инструментом проектирования компьютерных систем и их элементов. Вместе с тем отсутствие теоретической базы является главным недостатком архитектурных методов проектирования. С этим недостатком связана ограниченность количества характеристики сложности архитектурной модели, которые используются для синтеза и оптимизации аппаратно-программных средств. Оптимизация технических решений в процессе разработки ограничена лишь минимизацией временной и аппаратной сложности без учёта информационных характеристик. В статье описывается возможность расширения списка характеристик компьютерных средств на основе модели аппаратно реализуемого алгоритма.

Особенности алгоритмических проблем. Алгоритм – фундаментальное понятие математики, ему нельзя дать точное определение с привлечением других понятий. Понимание и использование

Черкасский М., д.т.н., профессор Национального университета «Львовская политехника», Украина.

Саченко А., д.т.н., профессор Тернопольского национального экономического университета, Украина.

таких понятий возможно посредством зримых моделей, которые помогают описать их сущность. Например, фундаментальному понятию евклидовой геометрии "прямая" можно привести аналогию с тонким лучом или пересечением двух плоскостей и записать его уравнение. Это уравнение является математической моделью, которую можно использовать в точных вычислительных операциях для решения практических задач. Таким образом, можно дать точное определение математической модели, но не самому понятию.

Особенностью понятия алгоритма является то, что оно не имеет зримого образа. В связи с этим история развития понятия "алгоритм" и разработка моделей была значительно сложнее, чем, например, фундаментальные понятия евклидовой геометрии. Первые эффективные алгоритмы были созданы еще до н.э.[5]. Революционное значение в упрощении расчетов имели работы Аль-Хорезми, где он впервые описал реализацию арифметических операций [6].

Необходимость в применении моделей алгоритма возникла в связи с исследованием проблем разрешимости. Сущность этих проблем заключалась не в построении алгоритма для решения некоторой задачи, а в доказание того, что задача не имеет решения. Поэтому и потребовалось уточнить интуитивное понятие алгоритма, для чего были созданы и математически точно определены формальные алгоритмические системы (ФАС), к которым относятся модели алгоритмов. Первая модель алгоритма была предложена Тьюрингом и независимо от него Постом в 1936 г., и приобрела название "машина Тьюринга". Эта машинно-ориентированная модель алгоритма, кроме проблем разрешимости, позволила исследовать принципы организации вычислительного процесса и конструирования компьютеров. Применение машины Тьюринга для решения вычислительных задач способствовало также разработке теории сложности алгоритмов.

Формальные алгоритмические системы. Формализация понятия "алгоритм" посредством исследования моделей алгоритма позволила избежать неоднозначности толкования алгоритма, определить перечень свойств, параметров и характеристик сложности. С помощью абстрактных моделей были уточнены следующие свойства алгоритма: дискретность, элементарность, детерминированность, массовость. В процессе разработки эффективных вычислительных процедур были задекларированы три характеристики алгоритмов и восемь его параметров. К набору взаимозависимых характеристик сложности относятся временная сложность – количество операций, необходимых для решения задачи избранным алгоритмом, емкостная сложность, количество ячеек памяти, количество команд программы. Иногда последнюю характеристику называют программной сложностью. В статье этот имеет другое содержание. Набор параметров постоянный для всех моделей алгоритма. Он включает: правило начала, систему начальных данных, правило ввода данных, правило непосредственной переработки, систему промежуточных результатов, систему конечных результатов, правило вывода результатов, правило окончания [5] (В Математической энциклопедии перечислено только 7 параметров, правило ввода ошибочно не упоминается). Изменение хотя бы одного параметра вызывает изменение значений характеристик сложности. Такая зависимость позволяет проводить параметрическую оптимизацию алгоритмов.

С точки зрения полноты свойств, характеристик, параметров, их взаимозависимости, простоты реализации вычислительного процесса машина Тьюринга является наиболее благоприятной среди других абстрактных моделей для разъяснения основных идей конструирования модели компьютерного алгоритма.

Особенности машины Тьюринга. Машина Тьюринга – это абстрактная модель ФАС, которая соответствует интуитивному толкованию алгоритма. Она имеет одну и ту же неизменную конфигурацию мысленно воображаемых средств реализации алгоритма, пригодной для решения любой вычислимой задачи. К воображаемым средствам относятся бесконечная неподвижная лента, подвижная головка и связи, которые соединяют головку с блоком программы. Программа является частью модели алгоритма. В явном виде в машине отмечены все восемь параметров. Особенности ее работы не противоречат свойствам алгоритма, её шаги дискретны и детерминированы, обладают свойством массовости. Единственное свойство,

которое не определяется формально точно, это элементарность шага алгоритма. В машине Тьюринга шаг алгоритма сопровождается несколькими операциями: чтением символа в ячейку ленты, поиском необходимой команды, выполнением команды по изменению содержания ячейки ленты (оставить предыдущий символ, стереть его, записать ли новый), операцией перемещения головки (оставить на месте, сдвинуть влево или вправо). Наборы операций одного шага могут изменяться. Таким образом, элементарность шага машины Тьюринга является условной, формально она не определена.

Формальная неопределенность свойства "элементарность", шага машины Тьюринга, как и любой другой абстрактной модели алгоритма, является принципиальным недостатком теории абстрактных алгоритмов.

Формальное определение детерминированной, одноленточной машины Тьюринга следующее: машины Тьюринга – это шестерка:

$$M = \langle A, Q, q_0, q_f, a_0, P \rangle, \quad (1)$$

где A - конечное множество символов внешнего алфавита;

Q - конечное множество символов внутреннего алфавита;

q_0, q_f - начальное и конечное состояния машины Тьюринга

$q_0, q_f \in Q$;

a_0 - обозначение пустой ячейки бесконечной ленты $a_0 \in A$;

P - такая программа, которая не может иметь двух команд, в которых совпадают два первых символа:

$$P = \{A\} \times \{Q\} \rightarrow \{A\} \times \{L, R, S\} \times \{Q\}, \quad (2)$$

где L - сдвигать головку влево; R - вправо; S - оставить на месте [7].

Каждый экземпляр машины Тьюринга моделирует один алгоритм, каждому алгоритму отвечает собственная программа. Минимизация временной или емкостной сложности при решении одной и той же задачи достигается изменением одного или нескольких параметров. Изменение конфигурации и "объема оборудования" машины в процессе вычислений не происходит. В формальном определении машины Тьюринга (1) не фигурируют аппаратные средства. Эти средства (головка, лента, связи с узлом программы) являются воображаемыми, они используются только для объяснения процесса реализации алгоритма. Собственно машина Тьюринга – это лишь воображаемый образ вычислителя, и декларируется в ней только программа. Модели ФАС, в определениях которых отсутствуют аппаратные средства, называются абстрактными моделями алгоритмов. Следствием этого является отсутствие в списке характеристик алгоритма аппаратной сложности.

Эта особенность машины Тьюринга и других абстрактных моделей алгоритма – неучтено влияние аппаратных средств на сложность вычислительного процесса - является их главным недостатком в сравнении с архитектурной моделью. Поэтому выводы теории абстрактных алгоритмов в процессе исследования и проектирования компьютерных средств в современных условиях практически не используются.

Аксиомы компьютерных алгоритмов. До появления первых компьютеров дискутировался вопрос о возможности построения теории алгоритмов без аксиом. Такая точка зрения обосновывалась успешными результатами исследований проблем разрешимости и математических парадоксов.

В условиях бурного развития компьютерных систем архитектурная модель практически полностью вытеснила абстрактные модели ФАС из области проектирования аппаратно-программных средств. Ситуация изменилась при оценке сложности вычислений с применением компьютеров. Теория абстрактных ФАС начала давать неверные результаты. Примером этого явилась теорема о линейном ускорении [8]. Для эффективного использования моделей алгоритмов в процессе разработки вычислительных средств стало необходимым устранение противоречий, присущих абстрактным ФАС.

Первое противоречие формального определения абстрактных ФАС заключается в следующем. Для объяснения процесса работы алгоритма абстрактные ФАС (машина Тьюринга, Нормальные алгоритмы Маркова, модель Колмогорова и другие) используют аппаратные средства,

но лишь для демонстрации процесса вычислений. В формальном определении модели, аппаратные средства не декларируются, и в оценке характеристик сложности алгоритма не используются.

Второе противоречие состоит в неполноте набора взаимосвязанных характеристик сложности абстрактных ФАС. Кроме аппаратной сложности в процессе анализа алгоритмов не оценивается его информационное содержание, та его составляющая, которая вносится разработчиком.

Эти противоречия устраняются уточнением основных понятий теории алгоритмов, в том числе самого понятия "алгоритм". Для этого необходимо построение новой аппаратно-программной модели алгоритма, базирующейся на аксиомах компьютерной теории алгоритмов [9, 10].

Аксиома I. Алгоритмы могут быть реализованы аппаратными средствами.

Аксиома II. Алгоритмы могут быть реализованы аппаратно-программными средствами.

Аксиома III. Алгоритмы не могут быть реализованы только программными средствами.

Сформулированные аксиомы не противоречат интуитивному толкованию понятие "алгоритм". Они подчёркивают необходимость при разработке новых моделей ФАС учитывать аппаратные средства. Примером такой аппаратно-программной модели является SH-модель алгоритма (Software/Hardware-model) [3].

Определение SH-модели. SH-модель – это семёрка

$$SH : \langle D, Q, q_0, q_f, G, P, M \rangle,$$

где D - конечное множество символов внешнего алфавита;

Q - конечное множество состояний SH-модели;

q_0, q_f - начальное и конечное состояния $q_0, q_f \in Q$;

G - конфигурация аппаратных средств модели;

P - программа, $P = \{y_i \mid i = 1, n\}$;

M - память.

$$G = (X, U),$$

где X - множество элементарных преобразователей;

U - множество меж-соединений.

Центральным моментом структурных построений SH-модели является элементарный преобразователь.

Элементарный преобразователь. SH-модель не имеет раз и навсегда установленной структуры аппаратных средств, подобной структуре машины Тьюринга. Однако каждая конкретная модель алгоритма в части аппаратного строения имеет точно очерченную структуру, которая состоит из двух непересекающихся множеств: множества элементарных преобразователей и множества меж-соединений [3]:

$$E = \{e_1, e_2 \dots e_n\};$$

$$U = \{u_1, u_2 \dots u_m\}.$$

Аппаратные средства содержат один или несколько элементарных преобразователей. Элементарный преобразователь x_i преобразует некоторую совокупность начальных данных d_i в совокупность выходных данных d'_i :

$$x_i : \{d_i\} \rightarrow \{d'_i\}.$$

Элементарный преобразователь – воображаемый образ неделимого элемента схемы аппаратных средств SH-модели, который задается черным ящиком, который имеет входы и выходы и реализует наперед заданную функцию. Будем различать два вида черных ящиков: а) с входом и выходом данных, б) с входом и выходом данных, а также со входом управления. Первой моделью является элементарный преобразователь с аппаратным преобразованием данных (аппаратная реализация алгоритма), второй является - элементарный пре-

образователь с аппаратным преобразованием данных с внешним управлением (аппаратно-программная реализация алгоритма).

Определение: а) элементарный преобразователь без внешнего управления - это тройка:

$$(I, \Phi, O),$$

где I - вход данных, Φ - функция преобразования, O – выход результатов преобразования данных.

б) элементарный преобразователь с внешним управлением – это четверка:

$$(I, \Phi, Y, O),$$

где I - вход данных, Φ - функция преобразования, Y – вход управляющего сигнала, O – выход результатов преобразования данных.

Исследование SH-модели позволяет сформулировать выводы, которые являются следствиями сформулированных аксиом и введенного понятия "элементарный преобразователь".

- **Элементарный преобразователь** является математическим объектом. Отсюда понятие аппаратная сложность также является математическим объектом.
- В перечень характеристик сложности модели компьютерного алгоритма вводится дополнительная в сравнении с абстрактным алгоритмом характеристика **аппаратная сложность**.
- Единицей аппаратной сложности является **элементарный преобразователь**, который задается черным ящиком. Черный ящик имеет две формы представления. Первая форма – без управления (отвечает аксиоме I), вторая форма – с управлением (отвечает аксиоме II). Элементарный преобразователь.
- Такое представление аппаратных и аппаратно-программных средств позволяет: 1) уточнить понятие свойства алгоритма **элементарность**, дать ему точное определение: элементарный шаг алгоритма – это операция преобразования, или передачи данных элементарным преобразователем 2) ввести новое свойство компьютерного алгоритма **иерархичность**: элементарный преобразователь является SH-моделью алгоритма следующего иерархического уровня аппаратных средств.
- Наличие множества элементарных преобразователей и множества связей между ними позволяет ввести принципиально новую характеристику компьютерного алгоритма – **структурную сложность**. Это **информационная характеристика**. Она является логарифмической мерой степени нерегулярности связей множества элементарных преобразователей (матрицы смежности элементарных преобразователей).
- Введена новая характеристика – **программная сложность**, которая также является информационной характеристикой. Это логарифмическая мера степени нерегулярности сигналов управления временной диаграммы работы аппаратно-программных средств.
- Программа не является компьютерным алгоритмом (этот вывод соответствует аксиоме III). Программа является составной частью компьютерного алгоритма.

Сравнение моделей. Таким образом, уточнение понятия "алгоритм" на основе SH-модели позволяет ввести в теорию алгоритмов принципиально новые понятия. Это относится к уточнению и формулировке свойств элементарность и иерархичность, а также к "легализации" аппаратных средств в определении ФАС. Все это позволяет ликвидировать разрыв между теорией и практикой исследования компьютерных систем. Технология проектирования получила фактически новую теорию сложности аппаратно-программных средств в сравнении с архитектурной моделью. Синтез, анализ и оптимизация компьютерных алгоритмов проводится по расширенному (количественно и качественно) списку взаимозависимых характеристик сложности на всех иерархических уровнях построения системы. В таблице сравниваются три рассмотренные модели по списку характеристик сложности моделей и списку свойств.

Модели Характеристики	Архитектурная модель	Абстрактные ФАС	Компьютерные ФАС
Временная сложность	+	+	+
Аппаратная сложность	+	-	+
Ёмкостная сложность	+	+	+
Программная сложность	-	-	+
Структурная сложность	-	-	+
Дискретность	+	+	+
Детерминированность	+	+	+
Элементарность	+,-	+,-	+
Массовость	+	+	+
Иерархичность	+	-	+

Как видно из данной таблицы, компьютерные ФАС имеют наиболее полные характеристики.

Выводы

1. Архитектурные методы проектирования компьютеров основаны на опыте разработчика, умении находить новые схемотехнические и системотехнические решения, которые только частично базируются на использовании характеристик сложности абстрактных алгоритмов.
2. Модели абстрактных формальных алгоритмических систем, которые созданы для исследования проблем разрешимости, не эффективны для разработки реальных аппаратно-программных средств, так как не учитывают влияние аппаратных средств на сложность вычислительного процесса.
3. Модель аппаратно-программных алгоритмов (SH-модель) с объявлением в своём определении аппаратно-программных средств является основой новой технологии проектирования компьютерных систем.
4. Центральным понятием аппаратных средств является "элементарный преобразователь", который позволяет уточнить понятие теории алгоритмов "элементарность" и "иерархичность".
5. Использование множества элементарных преобразователей позволило сформулировать понятие характеристики "аппарат-

ная сложность" и уточнить понятие характеристики "временная сложность".

6. Исследование компьютерных систем посредством теории сложности SH-модели алгоритма имеет преимущества в сравнении с методами архитектуры и методами анализа абстрактных формальных алгоритмических систем.
7. Следует различать программную и структурную сложность.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Трахтенброт Б.А. Алгоритмы и вычислительные автоматы. – М.: Сов. радио, 1974. – 200 с.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание. : Пер.с англ. – М.: Изд. дом «Вильямс», 2005. – 1296 с.
3. Черкаський М.В. SH-модель алгоритму // *Комп'ютерна інженерія та інформаційні технології. Вісник НУ "Львівська політехніка"* – Львів, 2001, №433, с.127-134.
4. Даль В.И. Толковый словарь живого великорусского языка в четырёх томах. – М.: изд "Русский язык" 1999.
5. Математическая энциклопедия. Гл. ред. И.М. Виноградов. – М.: Советская Энциклопедия, 1977. – в 5-и томах.
6. Мухаммед ибн Муса ал-Хорезми. Книга об индийском счете.- Ташкент, изд-во "Фан", 1973.
7. Успенский В.А., Семенов А.Л.. Теория алгоритмов: основные открытия и приложения. – М.: Наука Гл. ред. физ.-мат. лит., 1987. – 288 с.
8. Блюм М. Машино-независимая теория сложности рекурсивных функций// Библиотека "Кибернетического сборника". Проблемы математической логики. М.: Мир, 1970.
9. Cherkaskyy M.V. Computer algorithms: axioms and consequences // *Advanced Computer Systems and Networks: Design and Application. Proceedings of the 2nd International Conference ACSN – 2005. September 21-23, 2005, Lviv – Ukraine.* – Publishing House of Lviv Polytechnic National University, 2005, p.147.
10. Anatoly Sachenko, Mykola Cherkaskyy, Oleksandr Osolinskiy. Algorithm Conception Clarification Based on the SH-Model. Proceedings of IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 6-8 September 2007, Dortmund, Germany. – Pp. 300-303.

Материал поступил в редакцию 31.10.08

CHERKASSKIJ M., SACHENKO A. Technology of designing on the basis of the theory of complexity of computer algorithms

The lacks of designs of abstract models of algorithms are considered. The necessity of specification of concept "algorithm" is shown. The new model of algorithm (SH-model) intended for construction of effective technology of designing of computer means is described.

УДК 681.4

Дудкин А.А.

ОСНОВНЫЕ ПОДХОДЫ К РЕАЛИЗАЦИИ ОБРАБОТКИ ИЗОБРАЖЕНИЙ ТОПОЛОГИЧЕСКИХ СЛОЕВ ИНТЕГРАЛЬНЫХ СХЕМ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ ПРОЕКТИРОВАНИЯ И ПРОИЗВОДСТВА ЭЛЕКТРОННЫХ ИЗДЕЛИЙ

Введение. Системы компьютерного зрения (СКЗ) относятся к промышленным программным продуктам, существенной чертой которых является высокий уровень сложности: один разработчик практически не в состоянии охватить все аспекты такой системы. Системы подобного типа обычно имеют большое время жизни, большое количество пользователей оказывается в зависимости от их нормального функционирования. Это обуславливает необходимость индустриализации производства программной продукции – применения высокоэффективных технологий создания программного обеспечения (ПО).

Основные подходы к реализации алгоритмов обработки. Ниже приведены основные подходы, которые нашли воплощение в проектировании систем обработки изображений топологии инте-

гральных схем (ИС), которые в совокупности обеспечивают эффективную обработку топологии ИС.

Объектно-ориентированное проектирование (ООП). В основе ООП лежит представление о том, что программную систему необходимо проектировать как совокупность взаимодействующих друг с другом объектов, рассматривая каждый объект как экземпляр определенного класса, причем классы образуют иерархию. Объектно-ориентированный подход отражает топологию языков высокого уровня, таких как Smalltalk, Object Pascal, C++, CLOS и Ada. Объектная модель имеет ряд преимуществ: позволяет в полной мере использовать возможности объектных и объектно-ориентированных языков программирования; существенно повышает уровень унификации разработки и пригодность для повторного использования не только программ, но и проектов, что ведет к созданию среды разра-