

УДК 658.512.011.56:[681.3.06+519.8/85]

ПОИСКОВОЕ КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ

Костерин В.В.

Волгоградский государственный технический университет, Волгоград

Параллельное развитие теории программирования и теории проектирования сделало актуальным их системное исследование. Цель исследований, отраженных в статье, состояла в достижении позитивного дальнейшего взаимного проникновения этих теорий. Теория проектирования объектов перенесена в новую область – теорию программирования. Теория программирования позволила разработать методику массового обучения функциональному документированию и дедуктивному мышлению, как будущих программистов, так и специалистов непрограммирующих специальностей, например, работающих в областях техники, экономики и даже учащихся среднего образования.

Параллельное развитие теории программирования и теории проектирования сделало актуальным их системное исследование. Цель исследований, отраженных в статье, состояла в достижении позитивного дальнейшего взаимного проникновения этих теорий.

В настоящее время программирование трансформировалось в целую индустрию производства программных изделий. Основные должности программистов: техник-программист, инженер-программист (третьей, второй и первой категорий). Следовательно, профессиональный разработчик программных изделий должен владеть теорией проектирования, методами активизации мышления.

На ранних этапах принимаются проектные решения, в большой степени определяющие качество и стоимость будущих изделий. Методической основой этих исследований являются методы поиска технических решений, предназначенные для использования человеком и отражающие обобщенный опыт проектирования, методы формализации знаний об объектах проектирования и процедур работы с этими знаниями, опыт и знания проектировщиков.

Ранее и до настоящего времени большая часть нестандартных задач решалась человеком на интуитивном уровне методом проб и ошибок: последовательное выдвижение и рассмотрение идей. Главный недостаток метода проб и ошибок – это, во-первых, медленное генерирование новых идей, а во-вторых, отсутствие защиты от психологической инерции, то есть выдвижение идей тривиальных, обыденных, неоригинальных.

Следующим шагом в совершенствовании технологии явился переход к направленным методам поиска решений, которые базируются: на раскрытии и описании процесса решения; на представлении его в виде некоторого эвристического алгоритма. Направленность эвристических

методов – раскатать мышление, помочь поновому увидеть задачу, преодолеть стереотипы.

При успешном решении какой-либо творческой задачи, человек получает два результата: это само решение поставленной задачи и методический опыт, т.е. уяснение процесса решения данной конкретной задачи. Такие методические правила называют эвристическими приемами. Эвристический прием – способ разрешения определенного противоречия. Но проблема заключается в том, что решение одной задачи нельзя просто перенести на решение другой. Поэтому только после решения определенного числа задач у человека появлялся набор правил, указаний или приемов решения той или иной задачи.

Используя фонд эвристических приемов, Воинов Б.С. и Костерин В.В. успешно синтезировали ряд новых механизмов алгоритма поиска глобального экстремума функций многих переменных на сетке кода Грея [1, 3].

Пример использования метода эвристических приемов для создания алгоритмов описан в книге Д. Пойа [4]. Цитируемая часто, данная книга содержит фонд лишь из 14 эвристических приемов! Фонд из 20 эвристических приемов, полезных для генерации структуры объектно-ориентированных программ, опубликован в книге Б. Страуструпа [5].

Описанный же в учебнике [2], только укороченный фонд включает 200 эвристических приемов для программирования.

Методы аналогий является наиболее популярными методами для программистов. Суть одного из методов состоит в следующем. Совершенствуемую систему держат как бы в фокусе внимания и переносят на нее свойства других, не имеющих к ней никакого отношения программ из коллекции. При этом возникают необычные сочетания, которые стараются развить дальше.

Согласно проведенному опросу, большинство профессиональных программистов именно этим методом генерировали внешний облик своих программных систем и определили способы реализации многих функций программ.

Метод морфологических таблиц является простым и эффективным методом, особенно там, где необходимо найти большое число вариантов достижения цели. В последнее время он используется достаточно широко как средство развития творческого воображения. Примером морфологической таблицы может являться прайс-лист компьютерной фирмы. В прайс-листе содержится информация: о нескольких типах корпусов ЭВМ; о нескольких типах материнских плат; о нескольких типах процессоров и т.д. Каждая часть характеризуется техническими характеристиками и ценой. Не все варианты частей могут быть состыкованы между собой. Главное, что характеризует прайс-лист – это отсутствие критерия качества целого компьютера для конкретного пользователя. Глядя на прайс-лист надо синтезировать данный критерий и выбрать оптимальный состав частей. Как результат синтеза могут быть выявлены варианты построения компьютеров, ориентированных на различные категории пользователей.

Уже в 1983 г. была успешно применена морфологическая таблица для синтеза идей построения алгоритма нелинейного программирования поиска глобального экстремума функций многих переменных на сетке кода Грея [1, 3]. Алгоритмы нелинейного программирования предназначены для поиска экстремумов функций многих переменных. В методах прямого поиска экстремум выявляется путем расчета множества точек функции при аргументах, определяемых самим алгоритмом поиска. В учебнике [2] приведена данная морфологическая таблица, которая содержит классификационные признаки отдельных механизмов алгоритмов нелинейного программирования на уровне основных принципов. Интересно отметить, что число возможных реализаций алгоритмов нелинейного программирования по этой таблице составляет $N = 5 \cdot 6 \cdot 8 \cdot 5 \cdot 7 \cdot 7 \cdot 6 = 352800$, что значительно превышает число опубликованных методов (около 3000)!

Исходя из принципа аналогии, выдвинута гипотеза, о том, что данная методика может найти применение для создания ряда новых численных методов вычислительной математики.

Методы аналогий, эвристических приемов и морфологического синтеза зарекомендовали себя эффективными методами для генерации идей построения больших программ и программных комплексов. Апробация данных методов была

осуществлена на десятках проектах программных систем, выполненных как автором, так и студентами-дипломниками, работающих под руководством автора.

Инструкция пользования каким-либо устройством, описание бизнес-процесса, инструкция вообще или алгоритм программы являются описаниями функционирования. Хорошим функциональным описанием является описание: безошибочное, однозначное для читателя, краткое, суть которого понимается быстро. Согласно методике [2], хорошее функциональное описание составляется от общего к частному с использованием особых конструкций предложений – типовых элементов (типовых структур или просто структур), составляющих семантический скелет будущих инструкций. Главное преимущество изложенной методики состоит в однозначности соответствия функционального описания замыслу, что достигается как оправданной декомпозицией, так и исчерпывающим тестированием.

Оказалось, что обучение методике разработки описаний функционирования систем от общего к частному (составление инструкций вообще, описание бизнес-процессов) вполне доступно студентам второго курса специальности бухгалтерский учет, даже если они не изучали эту методику в курсе программирования. Более того, половина учеников девятого класса обычной школы вполне способна полностью освоить данный материал.

Авторы считают, что в данной работе новыми являются следующие положения и результаты:

1) теория проектирования объектов техники и, в частности, теория поискового конструирования перенесена в новую область – теорию программирования;

2) теория программирования позволила разработать методику массового обучения функциональному документированию и дедуктивному мышлению, как будущих программистов, так и специалистов непрограммирующих специальностей, например, работающих в областях техники, экономики и даже учащихся среднего образования.

СПИСОК ЛИТЕРАТУРЫ

1. Воинов Б.С. Информационные технологии и системы: Монография. В 2 кн.

Книга 1. Методология синтеза новых решений. - Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2001. - 404 с.

Книга 2. Прикладные системные исследования. - Нижний Новгород: Изд-во ННГУ им. Н.И. Лобачевского, 2001. - 272 с.

2. Технологии программирования: Учебник /В.А. Камаев, В.В. Костерин. - М.: Высш. шк., 2005,-359 с.

3. Костерин А.В., Костерин В.В. Алгоритм поиска глобального экстремума и его применение для многопараметрического синтеза конструкций СВЧ В кн. Вопросы релятивистской СВЧ

электроники (сб. науч. трудов). М.: РТИ АН СССР, 1983, с. 168-185.

4. Пойа Д. Математическое открытие. Решение задач / Д. Пойа.- М., Наука, 1976. -389 с.

5. Страуструп Б. Язык программирования С++/. В двух частях. Киев: ДиаСофт, 1998.

SEARCHING DESIGN AND PROGRAMMING

Kosterin V.V.

Volgograd state technical university, Volgograd

Parallel development of the theory of programming and the theory of designing has made actual their system research. The purpose of the researches stated in article, has consisted in achievement of positive further mutual penetration of these theories. The theory of designing is transferred in new area – the theory of programming. The theory of programming has allowed to develop a technique of mass training to documenting and deductive thinking, both the future programmers, and experts not being programmers, for example, working in the field of technical equipment, economy and even pupils of secondary education.