

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ВИЗУАЛИЗАТОРОВ АЛГОРИТМОВ НА ОСНОВЕ АВТОМАТНОГО ПОДХОДА

М.А.Казаков, А.А.Шалыто

Санкт-Петербургский государственный университет информационных технологий, механики и оптики

E-mail: kazakov@rain.ifmo.ru

Одним из методических приемов [1-4] при обучении программированию и дискретной математике является разработка визуализаторов классических вычислительных алгоритмов [5, 6].

До настоящего времени визуализаторы этого типа строились эвристически [7]. При таком подходе каждый алгоритм требовал своего осмысления реализации визуализатора. Другими словами, технология разработки визуализаторов этого класса отсутствовала, и основные успехи в этой области были педагогическими.

В работе [8] был предложен автоматный подход к написанию программ, который базируется на таком понятии, как «состояние». Стало ясно, что традиционное построение визуализаторов, основанное только на использовании входных и выходных переменных, а также флагов для реализации логики при построении визуализаторов крайне неэффективно. Это объясняется тем, что при таком подходе обычно не ясно, как к визуализируемому алгоритму формально присоединять основные составляющие визуализатора – иллюстрации и комментарии.

При использовании автоматного подхода это становится естественным – каждому состоянию сопоставляется иллюстрация и комментарий. Автоматный подход позволяет также использовать в визуализаторах анимацию.

Ниже перечислены этапы предлагаемой авторами технологии построения визуализаторов алгоритмов на основе автоматного подхода:

- Постановка задачи, которую решает визуализируемый алгоритм.
- Решение задачи в словесно-математической форме.
- Выбор визуализируемых переменных.
- Анализ алгоритма для визуализации. Анализируется решение с целью определения того, что и как отображать на экране.
- Алгоритм решения задачи.
- Реализация алгоритма на выбранном языке программирования. На этом шаге производится реализация алгоритма, его отладка и проверка работоспособности.
- Построение схемы алгоритма по программе.
- Преобразование схемы алгоритма в граф переходов автомата, реализующего алгоритм, который может быть как автоматом Мили, так и автоматом Мура [8, 9].
- Преобразование автомата, реализующего алгоритм, в автомат визуализации.
- Выбор состояний, в которых будет выполняться анимация.
- Замена каждого из выбранных состояний тремя состояниями.
- Использование автомата анимации.
- Обеспечение взаимодействия между преобразованным и анимационным автоматами.
- Выбор интерфейса визуализатора.
- Сопоставление иллюстраций и комментариев с состояниями автомата.
- Обеспечение выбора в каждом анимационном состоянии статического либо динамического изображения, зависящего не только от состояния основного автомата, но и от номера шага анимации.
- Архитектура программы визуализатора.
- Программная реализация визуализатора.

Использование предлагаемой технологии на ряде примеров показало ее эффективность. При этом отметим, что после формального построения автомата по тексту программы, написанной вручную, собственно визуализатор построить достаточно просто. Так, например, для такого алгоритма, как «дискретная задача о рюкзаке» [9] визуализатор был построен в течение двух часов, в то время как при традиционном подходе путь от классической программы до визуализатора занимает обычно несколько дней из-за «неформального движения по этому пути». Кроме того, так построенный визуализатор практически не потребовал отладки.

Рассмотренная технология не автоматизирована, что затрудняет ее использование для сложных алгоритмов. В настоящее время ведутся работы по ее совершенствованию, в том числе и путем автоматизации [10]. На сайте <http://is.ifmo.ru> создан раздел «Визуализаторы», который содержит ряд визуализаторов, построенных как по предлагаемой технологии, так и на основе автоматизированного подхода [11].

Литература

1. Интернет-школа программирования. <http://ips.ifmo.ru/>.
2. Казаков М.А., Мельничук О.П., Парфенов В.Г. Интернет-школа программирования в СПбГИТМО (ТУ). Реализация и внедрение // Всероссийская научно-методическая конференция «Телематика'2002». СПб.: 2002. С.308-309. (http://tm.ifmo.ru/tm2002/db/doc/get_thes.php?id=170).

3. Столяр С.Е., Осипова Т.Г., Крылов И.П., Столяр С.С. Об инструментальных средствах для курса информатики // II Всероссийская конференция «Компьютеры в образовании». СПб.: 1994. С. 123-125.
4. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования // Международная научно-методическая конференция «Телематика'2000» . СПб.: 2000. С.189-191.
5. Кормен Т., Лайзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦМНО, 2000. 960 с.
6. Кнут Д. Искусство программирования. Т. 1: Основные алгоритмы. М.: Вильямс, 2001. 712 с.
7. Дискретная математика: алгоритмы. <http://rain.ifmo.ru/cat/>.
8. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. 628 с.
9. Казаков М.А., Шалыто А.А. Использование автоматного программирования для реализации визуализаторов // Компьютерные инструменты в образовании. 2004. №2, с.19-33. http://is.ifmo.ru/works/art_vis.pdf.
10. Казаков М.А., Корнеев Г.А., Шалыто А.А. Разработка логики визуализаторов алгоритмов на основе конечных автоматов // Телекоммуникации и информатизация образования", 2003, №6, с.27-58 <http://is.ifmo.ru/download/vis.pdf>.
11. Корнеев Г.А., Шалыто А.А. Требования к визуализаторам алгоритмов, выполняемых на базе технологии Vizi. Версия 4.1. <http://is.ifmo.ru/vis/req/>.