

ВИЗУАЛИЗАТОРЫ АЛГОРИТМОВ КАК ОСНОВНОЙ ИНСТРУМЕНТ ТЕХНОЛОГИИ ПРЕПОДАВАНИЯ ДИСКРЕТНОЙ МАТЕМАТИКИ И ПРОГРАММИРОВАНИЯ

ALGORITHM VISUALISERS AS MAIN INSTRUMENT FOR TECHNOLOGY OF TRAINING IN DISCRETE MATHEMATICS AND SOFTWARE DESIGN

Г.А.Корнеев, В.Г.Парфенов, С.Е.Столяр, В.Н.Васильев

Санкт-Петербургский государственный институт точной механики и оптики (технический университет), Санкт-Петербург

Тел.: (812) 532-55-50, e-mail: ses@mail.ifmo.ru

Одними из основных объектов изучения в курсах информатики и дискретной математики являются алгоритмы обработки данных.

Формально выстроенный курс алгоритмики, опираясь на используемый математический аппарат, является, как правило, самодостаточным и вполне может обойтись без привлечения иллюстративного материала. Нередко так дело и обстоит, поскольку иллюстрации делают полиграфическую продукцию дороже.

Отказ от чрезмерной экономии позволяет снабдить издание рядом иллюстраций и примерами, на которых разбирается работа алгоритма. При этом автор учебника, естественно, вправе ждать от читателя определенных интеллектуальных усилий, сводя к разумному минимуму количество и степень детализации примеров, которые призваны поддержать видеорядом обсуждаемый алгоритм. В конечном счете, независимо от объема, но лишь в силу своей статичности, такой видеоряд все равно может оказаться недостаточным для полного понимания читателем тонкостей алгоритма.

Те же рассуждения актуальны в отношении дистанционного курса, выложенного для изучения на сайт.

Примерно так же обстоит дело и с лекционным курсом, поскольку лектор, объективно, ограничен временем собственно учебного процесса и, субъективно, не склонен растягивать время предваряющей лекцию подготовки видеоряда. В этих условиях нередкой является ситуация, когда описание алгоритма, даже сопровождаемое фиксированным набором примеров, не отвечает двум целям: быть понятным и быть понятым.

Возможность достижения первой цели, в основном, определяется квалификацией лектора (либо, соответственно, автора учебника) и затраченными им усилиями.

Достижению же второй цели, как показывает наш опыт, весьма способствуют т.н. динамические визуализаторы алгоритмов. Их применение:

- позволяет изучать работу алгоритма в пошаговом режиме, аналогичном режиму трассировки среды программирования;
- допускает, в ряде случаев, трассировку укрупненными шагами, игнорируя рутинную часть вычислительного процесса, что существенно, например, для переборных алгоритмов;
- дает возможность варьировать входные наборы данных для рассматриваемого алгоритма, применить алгоритм к данным, подготовленным не автором учебного курса, а самим студентом.

Существенной методической особенностью визуализаторов является возможность вернуться к предыдущим шагам алгоритма и вновь "прокрутить" процесс с любого места рассмотрения.

На конференции "Телематика'2000" мы утверждали [1], что применение программ-визуализаторов "как в очном, так и в дистанционном обучении весьма способствует улучшению понимания учащимися лекционного материала". Сейчас, год спустя, мы готовы пойти дальше и рассматривать визуализаторы в качестве основного инструмента при обучении алгоритмике.

В пользу такого заключения свидетельствует уже немалый накопленный опыт. Так, читаемый на кафедре компьютерных технологий нашего института курс дискретной математики (его слушают более 70 студентов первого и второго года обучения) поддерживается и сопровождается в лекционном процессе несколькими десятками визуализаторов алгоритмов. Значительную часть того же курса (80 лекционных часов) прослушали студенты групп подготовки одной из компьютерных фирм (75 слушателей). Наконец, фрагменты этого курса (16 часов), в порядке эксперимента, читались, также с демонстрацией визуализаторов, группе студентов (25 человек) других кафедр.

Во всех, без исключения, указанных аудиториях демонстрация визуализаторов и обсуждение тонкостей работы алгоритмов именно с помощью этого инструмента вызвали заметный интерес. Как правило, те же слушатели обращались после занятий с просьбой предоставить им программы-визуализаторы для дальнейшего, самостоятельного тренажа. Результаты проводившихся в течение последнего года тестирований и экзаменов показывают, что алгоритмической частью курса большая часть наших студентов овладела вполне успешно.

Но, помимо полезности визуализаторов алгоритмов как инструмента обучения, необходимо отметить и другой, в определенном смысле, "побочный" продукт технологии обучения.

Очевидно, что само написание программы-визуализатора является нетривиальной задачей и, во всяком случае, требует определенного времени от разработчика. По вполне понятным причинам, мы с самого начала пошли по пути привлечения наших студентов к процессу создания нужных для учебного курса программ.

Несколько удачных студенческих разработок обнадежили нас, и в 2000/01 учебном году мы привлекли уже большую часть наших студентов к этой работе. При этом, учитывая широту "рынка рабочей силы" (более 100 студентов и слушателей курсов), многие визуализаторы предлагалось, независимо, разрабатывать нескольким программистам. Это позволило, в конечном счете, отобрать лучшие варианты для последующего использования в лекционном процессе.

Каждый студент-разработчик, в итоге, не только досконально разбирается в особенностях порученной ему темы, но и, что не менее важно, получает полезный опыт, реализуя не просто рядовой учебный проект, а создавая необходимое приложение.

Требования к программному продукту, связанные со сложившимися представлениями о необходимом дизайне визуализатора (впрочем, здесь разработчик обладает некоторой свободой выбора), наборе управляющих кнопок и режимах обработки данных, диктуют программисту определенный набор средств. Но, как оказалось, возникает и обратная связь с лекционным курсом. Так, упомянутый выше режим возврата к предыдущим состояниям процесса, в визуализации работы алгоритма, можно реализовать двумя, альтернативными, способами. Либо прокрутить все необходимые шаги от начала до нужного места, либо использовать специальные структуры, т.н. динамические множества с сохранением версий. Второй вариант, актуальный в ряде задач обработки данных, естественно, становится предметом специального рассмотрения в лекционной части курса, а студенты на собственном опыте убеждаются в его практической целесообразности.

Отметим, что ответственность за прием от студентов-разработчиков их проектов и соответствующая "сертификация" готовых продуктов были возложены также на студента, правда, старшего курса. Такая нагрузка, безусловно, является не только хорошей школой для будущего менеджера проектов, но и целесообразна с точки зрения подготовки будущих преподавательских кадров.

В конечном счете, созданный студентом визуализатор, если качество программного продукта признается адекватным задаче, быстро находит свое практическое применение. Примеры подобных разработок мы представим вместе с настоящим докладом.

Литература

1. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования. // Телематика'2000. Международная научно-методическая конференция. – Санкт-Петербург, 29 мая – 1 июня 2000 г.: Тез. докл. – С.189-191.