

Анализ топологий визуальных нотаций для записи алгоритмов и программ

Дробушевич Л.Ф., Коных В.В.

Кафедра информационных систем управления, факультет прикладной математики и информатики
Белорусский государственный университет
Минск, Беларусь
E-mail: droblf@bsu.by; konakh@bsu.by

Аннотация. Проведен анализ визуальных нотаций для записи алгоритмов и программ. Рассмотрены две визуальные нотации: нотация Дракон-схем и Р-схем. Показаны их возможности. Обосновывается использование Р-схем как для записи алгоритмов, так и для записи программ по спецификации алгоритма.

Ключевые слова: визуальные нотации, блок-схема, Дракон-схема, Р-схема, графический редактор.

Использование новых информационных технологий в учебном процессе позволяет достигнуть нового качества знаний. Инновационная деятельность в сфере образования – это вызов и требование времени. Инновационные технологии в университете – это, прежде всего, условие повышения качества образования и возможности получения более высоких позиций конкурентоспособности вуза и его выпускника на рынке образовательных услуг. Сегодня качество образования – один из главных вопросов образовательной политики государства.

В докладе проводится сравнение и анализ классических и современных визуальных нотаций, используемых в учебном процессе, для записи алгоритмов и программ.

Одной из первых и, пожалуй, самой известной нотацией, используемой для визуализации алгоритмов, являются блок-схемы. Однако на сегодняшний день эта нотация для записи и анализа алгоритмов утратила свою актуальность по нескольким причинам:

- В блок-схемах далеко не всегда можно явно выделить базовые алгоритмические конструкции современных языков программирования, т.е. алгоритм, записанный в виде блок-схемы, придется изменять и преобразовывать в код программы.
- При составлении блок-схем надо внимательно следить за структурностью алгоритма, т.е. блок-схема произвольного алгоритма должна быть композицией схем основных алгоритмических конструкций, в противном случае программирование подобного алгоритма будет затруднено.
- Неизбежно приходится переводить алгоритм в текстовую форму программы.
- Обозримыми, следовательно, и понятными являются блок-схемы только для небольших алгоритмов.

Одним из выходов из сложившейся ситуации может быть использование для записи алгоритмов визуальных нотаций «высокого уровня», которые с помощью соответствующего инструмента преобразуют графический алгоритм в программу.

В докладе для сравнения рассматриваются две визуальные нотации. Обе нотации достаточно известны,

обе имеют инструментальную поддержку для записи графического алгоритма и перевода алгоритма в программу. Первая – это нотация Дракон-схем, описанная В. Паронджановым [1], вторая – Р-схемы, предложенные И. В. Вельбицким [2]. На Р-схемы к тому же принят стандарт ISO/IEC 8631, что немаловажно в современном образовательном процессе.

Проанализируем возможности этих нотаций на двух примерах. В качестве первого примера рассмотрим два вложенных цикла с проверкой условий в начале циклов (типа while).

На Дракон-схеме (рис. 1а) приведена запись алгоритма цикла типа while с дополнительной проверкой условия внутри цикла. Р-схема этого же алгоритма (рис. 1б) явно показывает два вложенных цикла с помощью специальной структуры.

Во втором примере используются вложенные циклы с проверкой условия в конце (типа repeat). Это может быть, например, алгоритм поиска.

Возможная запись алгоритма с помощью нотации Дракон-схем показана на рис. 2а, алгоритмически эквивалентная запись в виде Р-схемы – на рис. 2в. Видно, что Р-схема явно демонстрирует вложенные циклы с помощью двух обратных дуг.

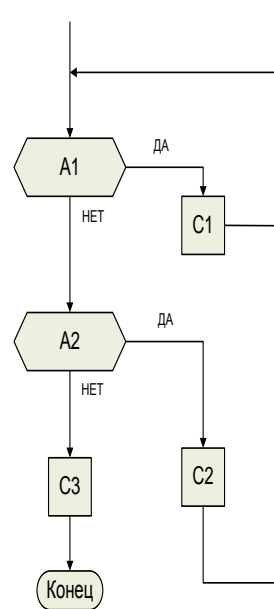


Рис. 1а

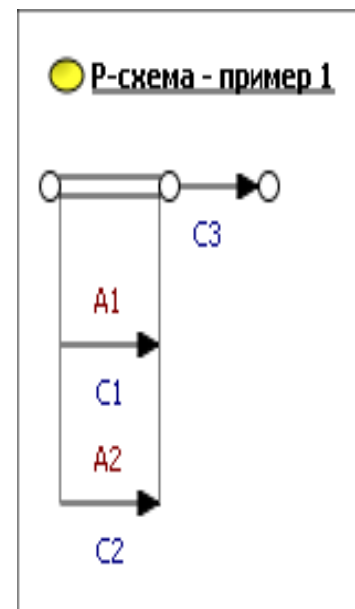


Рис. 1б

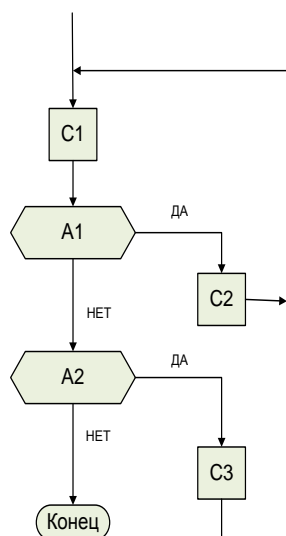


Рис. 2а

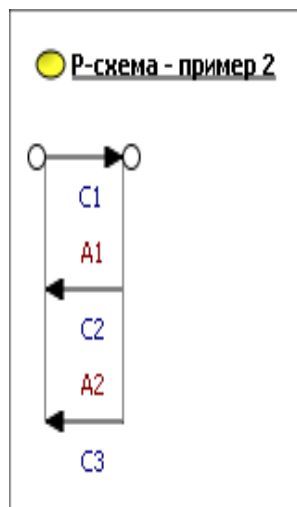


Рис.2б

Основные выводы:

- Оба примера наглядно демонстрируют, как не структурный алгоритм легко отображается в структурную P-схему согласно стандарту ISO/IEC 8631. Причем никаких семантических преобразований алгоритма при этом не пришлось делать.
- Топология P-схем более расширенная и управляемая по сравнению с топологией Дракон-схем, т.к. конструкции P-схем базируются на устойчивых формах.
- Представление алгоритма делает явной семантику различных ситуаций в поведении, а возможный маршрут выполнения алгоритма непосредственно связан с возможными семантическими состояниями системы.
- Графический язык P-схем специально сконструирован, чтобы осуществить плавный переход от алгоритма, представленного в виде двумерной картинки, к программе. При этом базис алгоритма и программы не изменяется. И таким базисом является P-схема.
- Преимуществом визуальной P-схемы является возможность быстрого ответа на вопрос, какими маршрутами может прийти управление в некоторую точку алгоритма, какие варианты поведения системы возможны после этой точки, сколько всего маршрутов. Это свойство P-схем можно эффективно использовать в процессе обучения для проведения инспекций и тестирования алгоритмов и программ.

В заключение можно сделать вывод, что P-схемы целесообразно использовать изначально в процессе обучения программированию, как для записи алгоритмов (спецификация алгоритмов), так и для записи программ по спецификации алгоритма [3]. Поэтому желательно P-схемы изучать и использовать параллельно с языками программирования.

Визуальное программирование существенно отличается от привычного текстового. Поэтому переход в процессе обучения на новую технологию связан, прежде

всего, с преодолением психологических трудностей у студента: эта смена мировоззрения, а не просто изучение новых редакторов и языков программирования.

Графический редактор должен «уметь», как минимум, по P-схеме генерировать структурированный код (прямое программирование) и обратно – по исходному коду генерировать P-схемы (обратное проектирование). Этот момент очень важен при обучении. И только в этом случае редактор можно включать в среду современной системы программирования или в инструмент моделирования программных систем. При таком подходе использование визуальной техники в практике программирования будет действительно эффективным. В противном случае будут потеряны достоинства визуального представления алгоритма.

Когда предполагается использование P-схемы, очень важно задать вопрос, зачем это делать, и как это поможет, когда дело дойдет до написания кода программы. Очевидного доказательства того, хорошие это методы или плохие, нет. Однако можно назвать причины, по которым лучше прибегнуть к использованию P-схем. Например, не вызывают сомнений следующие причины:

Появляется дополнительный уровень, на котором может быть проведено инспектирование алгоритмов на предмет их корректности независимо от нагромождений языка программирования, что повышает качество и надежность продукта.

Исключается возможность ошибок при переводе P-схемы алгоритма в программный код.

Появилась возможность визуально, понятно и достаточно точно представлять алгоритмы реализуемых методов классов.

Появляется возможность накапливать статистику по проценту ошибок в P-схемах, а затем использовать эти данные для предсказания количества дефектов в конечном продукте и характеристик его качества (сегодня системы качества продуктов актуальны).

Повышается дисциплина детального проектирования, что особенно важно в деле формирования профессиональной культуры в процессе обучения студентов.

Главное достоинство предлагаемой методики обучения студентов программированию – сквозная визуальная разработка от спецификации требований, системного анализа и проектирования до спецификации алгоритмов методов классов и далее – реализации тех же методов на соответствующем языке программирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] В. Д. Паронджанов Как улучшить работу ума: Алгоритмы без программистов – это очень просто! – М.: Дело, 2001.
- [2] И.В. Вельбицкий. Визуальная технология программирования нового поколения для широкого применения на базе стандарта ISO/IEC 8631 //Межд. конф. Кипр, 2010 г.
- [3] Л. Ф., Дробушевич, В. В. Конах Совместное использование визуальных нотаций UML и P-схем в процессе обучения объектно-ориентированным методам программирования // VI Межд. научн. конф. «Информационные системы и технологии» Минск, 2010.