

ИСПОЛЬЗОВАНИЕ ВИЗУАЛЬНЫХ ТЕХНОЛОГИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Дробушевич Л.Ф., Конах В.В.

Белорусский государственный университет

Минск, Беларусь

E-mail: droblf@bsu.by; konakh@bsu.by

Аннотация. Рассматриваются способы и методика использования визуальной технологии программирования нового поколения в процессе обучения профессиональному программированию. Определяются новые аспекты эффективного совместного использования Р-схем и UML в процессе обучения технологии объектно-ориентированного программирования. Дается ссылка на сайт Фонда Глушкова, где размещены открытые коды и Инструкция работы с Графическим редактором – основным ядром технологии.

Ключевые слова: визуальная технология нового поколения, графический редактор, Р-схема, сквозная визуализация процесса разработки программ.

Информатика как наука о способах обработки, хранения и передачи информации в настоящее время переживает ни с чем не сравнимый скачок в своем развитии, ибо практически все существующие отрасли человеческой деятельности в той или иной степени зависят от качества используемых информационных технологий.

Одной из основных целей вузовского курса информатики является обучение студентов решению задач сбора, преобразования, передачи и хранения информации, значимых с точки зрения профессиональной последующей деятельности.

Для решения сложных современных задач необходимо научить студентов разрабатывать соответствующее программное обеспечение. В связи с этим разделы, посвященные изучению программирования, являются важной неотъемлемой частью существующих курсов информатики. Изучение программирования, как правило, преследует две основные цели – выработку алгоритмического мышления и формирование навыков решения конкретных задач по обработке информации.

В последние годы в преподавании предмета «Информатика» в средней школе наметились определенные проблемы, которые позднее проявляются в вузе.

Суть основной проблемы заключается в тенденции понижения роли алгоритмизации и программирования в современном курсе школьной информатики. Основной упор в школьной программе делается на обучение пользовательским навыкам. В итоге большинство выпускников школы не получают необходимого уровня подготовки, у них не развивается пространственное и логическое мышление на том уровне, который необходим для вуза, отсутствуют навыки построения моделей и алгоритмов, им трудно осваивать вузовский курс программирования в полном объеме.

Использование визуальной технологии программирования на начальном этапе обучения программированию позволяет студенту достаточно быстро научиться разрабатывать качественные программы.

Изучение визуальной технологии не занимает много времени [1].

В этой технологии в качестве интернационального языка понимания любыми специалистами на всех этапах работы используется нагруженный по дугам граф, который изображается только горизонтальными и вертикальными линиями (дугами) трех типов – стрелки, направленные вправо, влево и специальные, изображаемые двойной линией без стрелки.

Нагрузки на графах записываются над дугой и под дугой в одну или несколько строк. Запись над дугой имеет смысл условия (Predicate) прохождения по дуге, а под дугой – выполняемого при прохождении по дуге по стрелке действия (Statement). Если условие ложно, то действие под дугой не выполняется, перехода по соответствующей дуге не производится. Условия и/или действия могут отсутствовать на дуге, что соответствует безусловному прохождению по дуге в направлении стрелки с выполнением или не выполнением действий. Вертикальные линии – вспомогательные, без стрелок и нагрузок и служат для соединения основных дуг с вершинами. Такие графы называются Р-схемами [1]. На графическую систему визуальной технологии имеются стандарты: ГОСТ 19.005-85 и ISO/IEC 8631. Если вертикальная линия соединяет две Р-схемы (их начальные и конечные вершины), то эти Р-схемы соответствуют изображению параллельно выполняемых процессов. При помощи Р-схемы можно описать любой алгоритм, программу и процесс их разработки. Примеры записи различных Р-схем приведены на рис.1.

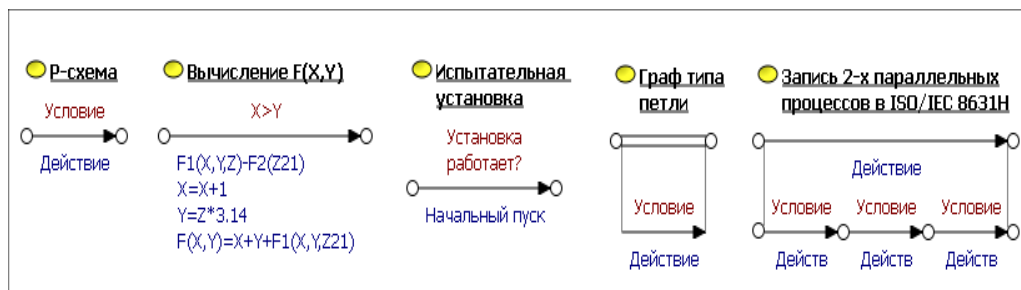


Рис.1. Примеры Р-схем.

Для ввода и редактирования Р-схем реализован Графический редактор, как ядро визуальной технологии нового поколения. На сайте (www.glushkov.org) приведены исходные коды Графического редактора, система его запуска и Инструкция пользователю. Этот редактор позволяет из Р-схемы получить программу на языке программирования, например, на C++.

Переход от привычных блок-схем, изучаемых в школьном курсе информатики, к Р-схемам не составляет большого труда.

Основное преимущество предлагаемого подхода заключается в том, что у студента появляется возможность без каких-либо трудозатрат перейти от описания спецификации задачи и алгоритма ее решения на естественном языке к формулировке предикатов и операторов языка программирования и получению качественной программы.

Второе преимущество предлагаемого подхода состоит в том, что появляется возможность использования навыков сквозной визуализации в процессе обучения моделированию программных систем (ПС). Модели полезны для понимания проблем, общения со всеми участниками проекта, для специфицирования предметной области, для проектирования ПО и баз данных [2]. Основным современным средством для представления таких моделей является нотация UML

(Унифицированный Язык Моделирования) [2,3]. Основной тезис использования в процессе обучения сквозной визуализации: «Р-схемы – не вместо нотации UML, а вместе и во взаимодействии!».

Методика обучения технологии объектно-ориентированного программирования, принятая в Белорусском государственном университете, отражает общепринятый в современной практике взгляд на то, каким образом графическую нотацию, процесс и инструментальное средство можно эффективно использовать совместно для создания не только модели проекта программной системы, но и для детального проектирования [3]. Эффективное совместное использование UML и Р-схем методологически возможно в *трех аспектах: дополнительно, совместно и вместо*.

1. Р-схемы дополнительно к диаграммам UML используются на этапе детального проектирования для спецификации алгоритмов функций системы и методов классов. Современные инструменты, поддерживающие процесс визуального моделирования с помощью нотации UML, позволяют получать по элементам модели каркасы исходных кодов на любых объектно-ориентированных языках программирования. Однако ни один инструмент не поддерживает визуализацию на этапе реализации проекта (реализация функций или методов классов разрабатываемой системы). Предлагается на этапе реализации проекта использовать Р-схемы в качестве визуального средства построения кода, т.е. логику реализации функций представлять в виде Р-схем. Далее – встроенный в среду проектирования (или моделирования) графический редактор автоматически переведет Р-схемы в традиционные исходные коды на языке программирования.

Такой подход позволяет в процессе написания кода больше размышлять над логикой алгоритма, а не над правилами и синтаксисом языка программирования.

Решение об использовании Р-схем в процессе детального проектирования принимается в зависимости от каждого конкретного приложения. Р-схемы могут оказаться очень полезными, если не для всего приложения, то для отдельных сложных и ответственных его частей, которые должны быть качественными и надежными.

Использование сквозной визуализации в процессе разработки программных проектов позволит повысить качество и надежность программ т.к. на этапе кодирования программисту не придется заниматься мелкими деталями. При этом основной *механизм абстрагирования* объектно-ориентированного подхода будет активно использоваться разработчиком на всех этапах разработки программного проекта: от эскизного до детального проекта и далее – кода реализации.

Такой подход позволяет работать на более высоком уровне абстракции и обеспечивает *простой и бесшовный* переход от проекта к коду.

Одним из преимуществ этого подхода является то, что разработчик может *проинспектировать алгоритм* отдельно от программы без углубления в сложности и детали языка программирования. Благодаря этому некоторые дефекты могут быть обнаружены до того, как они перерастут в дефекты программного кода. Методы со сложными ветвлениями и циклами являются первыми кандидатами для построения Р-схем. Инспектирование до программирования – основная идея использования Р-схем на данном этапе разработки.

Второе преимущество данного подхода состоит в том, что можно обеспечить *функцию поддержки кода и модели* алгоритма в актуальном состоянии. Причем такая функция может работать в реальном масштабе времени.

Создав Р-схему алгоритма, можно тут же сгенерировать по ней исходный код со всеми деталями и тонкостями языка программирования. И, наоборот, имея исходный код, можно получить по нему Р-схему алгоритма. Причем, если реализовать (в будущем) в редакторе режим синхронизации, инструмент будет делать это автоматически. Визуальная схема алгоритма реализованного метода в дальнейшем может оказать разработчику большую помощь в процессе тестирования и отладки программ.

Таким образом, благодаря использованию новой несложной в понимании и записи графической нотации Р-схем на этапе детального проектирования:

- появилась возможность визуально, понятно и достаточно точно представлять алгоритмы реализуемых методов классов свободных функций (утилит);

- исключается возможность ошибок при переводе Р-схемы алгоритма в программный код;

- появляется дополнительный уровень, на котором может быть проведено инспектирование алгоритмов на предмет их корректности независимо от языка программирования, что повышает качество и надежность продукта;

- появляется возможность накапливать статистику по проценту ошибок в Р-схемах, а затем использовать эти данные для предсказания количества дефектов в конечном продукте и характеристик его качества;

- повышается дисциплина в ходе детального проектирования, что важно в деле формирования профессиональной культуры в процессе обучения студентов.

2. Р-схемы совместно с диаграммами взаимодействий объектов (Sequence Diagrams, Communication Diagrams) на этапе анализа и специфицирования поведения выделенных прецедентов или вариантов использования системы (use cases).

Характеристики поведения разрабатываемой системы фиксируются и документируются в UML средствами модели вариантов использования, которая отображает все основные пользовательские функции системы. Такая модель особенно важна при организации и моделировании поведения всей системы и документирования *сценариев вариантов использования* (особенно, когда они имеют ветвления и циклы).

В рамках UML термин «сценарий» означает одну из возможных ветвей варианта использования, определяющую некоторую конкретную комбинацию условий. В современной практике программирования специфицирование сценариев варианта использования делается, как правило, в текстовой или табличной форме. В описание должны входить, как минимум, следующие разделы: *краткое описание* варианта использования; *основной поток* событий и *подчиненные* ему потоки; *альтернативные потоки*, связанные с обработкой ошибочных и исключительных действий пользователей.

Как показала практика обучения моделированию ПС вместо текстовых описаний основных, подчиненных и альтернативных потоков эффективнее использовать визуальную форму в виде Р-схем. Нагрузки на дугах Р-схемы могут быть дополнительно специфицированы примечаниями с текстом, которые в последующем могут стать прототипами операций и методов совместно с атрибутами.

Особенно полезно визуальное специфицирование потоков варианта использования непосредственно перед *моделированием реализации* этих потоков в виде диаграмм UML, которые отражают взаимодействие объектов друг с другом и которые получили название *диаграмм Взаимодействий*. Они бывают двух видов:

диаграммы Последовательностей (Sequence Diagrams) и диаграммы Коммуникаций (Communication Diagrams). Оба вида диаграмм служат для моделирования поведения объектов, т. е. с их помощью можно смоделировать динамические аспекты системы.

Чаще всего на практике с помощью диаграмм Взаимодействий моделируют последовательные неветвящиеся потоки управления, хотя в UML есть средства для моделирования итераций и ветвлений. Совместное использование на этом этапе моделирования Р-схем и диаграмм UML поможет при моделировании сложных потоков, содержащих итерации и ветвления.

3. Р-схемы вместо диаграмм Активности (Activity diagram) там, где их использование целесообразно в процессе моделирования системы.

Диаграмма Активности (или диаграмма Деятельностей) – диаграмма UML на первый взгляд напоминает привычную всем блок-схему, отображающую *алгоритм*.

Деятельности – это узлы, включающие действия и потоки управления и/или данные между действиями. Диаграммы Деятельностей демонстрируют последовательность действий (чаще всего методы классов), потоки управления, точки принятия и объединения решений. Поэтому такие диаграммы можно использовать для моделирования *динамики рабочих процессов (workflow)* и *потоков управления*, как всей системы, так и отдельных ее частей и далее операций.

Вообще говоря, *любой элемент* модели (класс, интерфейс, вариант использования, операция и др.), имеющий динамическое поведение, может быть дополнен диаграммой Активности – именно для уточнения динамики (в качестве примера можно применить диаграммы Активности для описания бизнес-процессов).

Диаграммы Деятельностей также поддерживают параллельные процессы, что немаловажно для исключения избыточных процессов при моделировании.

Но создавать и редактировать диаграммы Деятельностей как показывает практика их использования в процессе обучения, не совсем просто в традиционных инструментах, поддерживающих UML. По этой причине студенты да и разработчики программных проектов используют их в процессе моделирования довольно редко. Поэтому целесообразнее *вместо* диаграмм Деятельностей использовать Р-схемы (их проще рисовать, редактировать и понимать).

Заключение.

Главное достоинство предлагаемой методики обучения студентов специальности информатика – *сквозная визуальная разработка* от спецификации требований, системного анализа и проектирования до спецификации алгоритмов методов классов и далее – реализации тех же методов.

ЛИТЕРАТУРА

1. *Вельбицкий, И. В.* Визуальная технология программирования нового поколения для широкого применения на базе стандарта ISO/IEC 8631 / И. В. Вельбицкий // Межд. конф. MEDIAS - 2010, Кипр, 2010.
2. *Буч, Г.* Язык UML: Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон ; пер. с англ. М., 2000. 432с.
3. *Дробушевич, Л.* Визуальное моделирование программных систем с помощью Enterprise Architect 7.0 и UML / Л. Дробушевич, А. Исаченко. Минск: БГУ, 2009. 160 с.
4. *Фаулер, М.* UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фаулер, К. Скотт ; пер с англ. М., 1999. 191с.