Шалыто А.А.

ТЕХНОЛОГИЯ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ[1]

Санкт-Петербургский государственный институт точной механики и оптики (технический университет), Санкт-Петербург, shalyto@mail.ifmo.ru

Фундаментальной научной проблемой в области программирования является создание технологии, поддерживающей такие этапы создания программного обеспечения как проектирование, реализация, отладка и документирование. Несмотря на успехи в разработке технологий для поддержки отдельных этапов жизненного цикла программного обеспечения до настоящего времени отсутствует технология, охватывающая все указанные выше этапы [1,2].

В 1991 г. автором был предложен подход к созданию программного обеспечения, названный "автоматное программирование" [3]. Если в программировании в последнее время все шире используется понятие "событие", то предлагаемый подход базируется на понятии "состояние". Добавляя к нему понятие "входное воздействие", которое может быть входной переменной или событием, вводится термин "автомат без выхода". Добавляя к последнему понятие "выходное воздействие", вводится термин "автомат" (конечный, детерминированный).

При этом, область программирования, базирующаяся на этом понятии, может быть названа "автоматное программирование", а процесс создания таких программ — "автоматное проектирование программ".

Особенность рассматриваемого подхода состоит в том, что при его использовании автоматы задаются графами переходов, для различения вершин в которых вводится понятие "кодирование состояний". При выборе "многозначного кодирования" состояний с помощью одной переменной можно различить состояния, число которых совпадает со значностью выбранной переменной. Это позволило ввести в программирование такой термин, как "наблюдаемость программ".

В рамках предлагаемого подхода программирование выполняется "через состояния", а не "через переменные" (флаги), что позволяет лучше понять и специфицировать задачу и ее составные части.

В силу того, что в рамках автоматного подхода от графа переходов к тексту программы предлагается переходить формально и изоморфно, то при использовании языков программирования высокого уровня это наиболее рационально выполнять с помощью конструкции switch языка Си либо ее аналогов в других языках программирования. Последнее явилось основой того, чтобы назвать технологию автоматного программирования "Switch-технология".

В настоящее время эта технология разрабатывается в нескольких вариантах, различающихся как классом решаемых задач, так и типом вычислительных устройств, на которых осуществляется программирование.

В 1996 г. Российский фонд фундаментальных исследований (РФФИ) в рамках издательского проекта № 96-01-14066 поддержал издание работы [3], в которой предлагаемая технология была изложена применительно к системам логического управления, в которых события отсутствуют, выходные воздействия являются двоичными переменными, а операционная система работает в режиме сканирования. Системы этого класса реализуются обычно на программируемых логических контроллерах, которые имеют относительно небольшой объем памяти, а их программирование выполняется на таких специфических языках, как, например, язык функциональных блоков [4].

В дальнейшем автоматный подход был при участии автора распространен на событийные системы, которые называются также "реактивными" системами [5]. В системах этого класса указанные выше ограничения сняты. Как следует из названия этого класса систем, в них среди входных воздействий используются события, в качестве выходных воздействий — произвольные процедуры, а в качестве операционных систем — любые операционные системы реального времени.

Для программирования событийных систем с применением автоматов был использован процедурный подход, и поэтому такое программирование было названо "программирование с явным выделением состояний" [6].

При этом выходные воздействия "привязаны" к дугам, петлям или вершинам графов переходов (применяются смешанные автоматы — автоматы Мура-Мили), что позволяет в компактном виде представлять большое число последовательностей действий, которые являются реакциями на соответствующие входные воздействия.

Особенность предлагаемого подхода к программированию этого класса систем состоит в том, в них повышается централизация логики за счет устранения ее в обработчиках событий и формирования системы взаимосвязанных автоматов, которые вызываются из обработчиков [7]. Автоматы в системе могут взаимодействовать по вложенности, вызываемости и за счет обмена номерами состояний. Система взаимосвязанных автоматов образует системонезависимую часть программы, в то время системозависимую часть образуют функции входных и выходных воздействий, обработчики событий и т.д. Другая важнейшая особенность подхода состоит в том, что при его применении автоматы используются триедино: при спецификации; при программировании (сохраняются в программном коде); при протоколировании, выполняемом в терминах автоматов. Последнее позволяет контролировать правильность функционирования

[1] Работа выполнена при поддержке РФФИ по гранту № 02-07-90114 "Разработка технологии автоматного программирования".

системы автоматов. Протоколирование выполняется автоматически по построенной программе и может использоваться для задач большой размерности при сложной логике программы. При этом каждый построенный протокол может рассматриваться в качестве соответствующего сценария. Для таких задач невозможно применение диаграмм последовательностей и диаграмм кооперации, входящих в состав языка UML [8], так как их предлагается строить вручную при проектировании программы.

Подход предлагается применять не только при создании системы управления, но при моделировании объектов управления.

Он был апробирован при создании системы управления судовыми дизель-генераторами [9]. Эта система была специфицирована более, чем тридцатью взаимодействующими автоматами. Для описании модели дизеля также использовались автоматы. При проектировании на каждый автомат выпускалось четыре документа: словесное описание ("декларация о намерениях"), схема связей, поясняющая на русском языке символы, входящие в интерфейс автомата; граф переходов с символьными обозначениями событий, входных переменных и выходных воздействий; текст программного модуля, который реализует граф переходов (также без использования смысловых идентификаторов и комментариев). Эти документы заменяют самодокументирующиеся программы, содержащие смысловые идентификаторы и комментарии, так как такие программы при сложной логике понять не удается [10]. Эту проблему при сложной логике не решают и самодокументирующиеся графы переходов [8].

Для решения широкого круга задач весьма эффективен подход, основанный на совместном использовании объектной и автоматной парадигм, который в работе [11] был назван "объектно-ориентированное программирование с явным выделением состояний". Особенности этого подхода состоят в следующем. Также как и в машине Тьюринга [12] явно выделены управляющие (автоматные) состояния объекта, число которых значительно меньше числа остальных состояний, например, "вычислительных". В программирование введено понятие "пространство состояний", под которым понимается множество управляющих состояний объекта. Это обеспечивает существенно более понятное поведение по сравнению со случаем, когда такое пространство или ориентация в нем отсутствует). Предложен минимальный набор документов, позволяющий наглядно и строго описывать как структурные (статические), так и поведенческие (динамические) стороны программ.

Как и при использовании любого другого подхода, применение предлагаемого связано с множеством эвристик, возвратов назад, уточнений и параллельно выполняемых работ. Однако после завершения создания программы предлагаемый подход может быть сформулирован (по крайней мере для ее документирования) как "идеальная" технология, фиксирующая принятые решения. На основе анализа предметной области выделяются классы, и строится диаграмма классов. Для каждого класса разрабатывается словесное описание, по крайней мере, в форме перечня решаемых задач; для каждого класса создается структурная схема, отражающая его интерфейс и структуру. При этом атрибуты и методы разделены на автоматные и остальные. При наличии в классе нескольких автоматов строится схема их взаимодействия. Для каждого автомата разрабатываются словесное описание, схема связей, граф переходов. Каждый класс реализуется соответствующим модулем программы. Его структура должна быть изоморфна структуре класса, а методы, соответствующие автоматам, реализованы по шаблону, приведенному в работе [7]. Для отладки системы и подтверждения правильности ее работы автоматически строятся протоколы, в которых описывается функционирование объектов, содержащих автоматы, в терминах состояний, переходов, событий, входных и выходных воздействий. Выпускается проектная документация, составной частью которой является документация на программу.

Из изложенного следует, что применение автоматов проясняет поведение программы, так же как использование объектов проясняет ее структуру [13].

В описанной технологии автоматы применялись как методы классов, в рамках этой технологии могут быть использованы и другие подходы к объектной реализации автоматов, изложенные, например, в работах [14-16]. Особенности автоматной реализации параллельных процессов на основе механизма обмена сообщениями рассмотрены в работе [17].

27 ноября 2002 г. на открытии полуфинальных соревнований командного чемпионата мира по программированию ACM (Северо-Восточный Европейский регион) нами было объявлено об организации "Движения за открытую проектную документацию". В рамках этого движения на сайте http://is.ifmo.ru создан раздел "Проекты", в котором в ближайшее время будет размещено около 40 проектов разработки программного обеспечения на основе автоматного подхода.

Автоматный подход используется в настоящее время и при реализации вычислительных алгоритмов [18-20]. На основе автоматов предложен новый подход к построению визуализаторов алгоритмов, используемых на кафедре "Компьютерных технологий" при обучении программированию и дискретной математике [21].

Одна из целей настоящей работы состоит в том, чтобы показать, что автоматы в программировании служат не только "для распознавания цепочек символов" и управления "стиральными машинами", а также то, что они являются не просто одной из математических моделей дискретной математики, а могут применяться при реализации любых программ, обладающих сложным поведением.

По сравнению с известными технологиями программирования, в которых применяются автоматы [5,8], предлагаемая технология предназначена для более широкого класса вычислительных устройств и охватывает все этапы создания программ.

Литература

1. Шалыто А.А. Алгоритмизация и программирование для систем логического управления и "реактивных" систем. Обзор //Автоматика и телемеханика. 2001. №1.
2. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000.
3. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
4. Шалыто А.А. Реализация алгоритмов логического управления программами на языке функциональных блоков //Промышленные АСУ и контроллеры. 2000. №4.
5. Harel D., Politi M. Modeling Reactive Systems with Statecharts. NY: McGraw-Hill, 1998.
6. Шалыто А., Туккель Н. Программирование с явным выделением состояний //Мир ПК. 2001. №8,9.
7. Шалыто А.А., Туккель Н.И. SWITCH-технология — автоматный подход к созданию программного обеспечения "реактивных" систем //Программирование. 2001. №5.
8. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. М., ДМК, 2000.
9. Шалыто А.А., Туккель Н.И. Проектирование программного обеспечения системы управления дизель-генераторами на основе автоматного подхода //Системы управления и обработки информации. 2003. Вып.5. СПб.: ФГУП "НПО "Аврора".
10. Безруков Н. Повторный взгляд на "собор" и "базар" //BYTE/Россия. 2000. №8.
11. Шалыто А.А., Туккель Н.И. Объектно-ориентированное программирование с явным выделением состояний //Материалы международной научно-технической конференции "Искусственный интеллект - 2002". Т.1. Таганрог - Донецк: ТГРУ - ДИПИИ, 2002.
12. Шалыто А., Туккель Н. От тьюрингова программирования к автоматному //Мир ПК. 2002. №2.
13. Шалыто А.А., Туккель Н.И. Танки и автоматы //BYTE/Россия. 2003. № 2.
14. Гуров В.С., Нарвский А.С., Шалыто А.А. Автоматизация проектирования событийных объектно-ориентированных программ с явным выделением состояний //Труды X Всероссийской научно-методической конференции "Телематика-2003". Т.1. СПб.: СПбГИТМО (ТУ), 2003. http://tm.ifmo.ru
15. Шопырин Д.Г., Шалыто А.А. Применение класса "STATE" в объектно-ориентированном программировании с явным выделением состояний //Труды X Всероссийской научно-методической конференции "Телематика-2003". Т.1. СПб.: СПбГИТМО (ТУ), 2003. http://tm.ifmo.ru
16. Корнеев Г.А., Шалыто А.А. Реализация конечных автоматов с использованием объектно-ориентированного программирования //Труды X Всероссийской научно-методической конференции "Телематика-2003". Т.2. СПб.: СПбГИТМО (ТУ), 2003. http://tm.ifmo.ru
17. Гуисов М.И., Кузнецов А.Б., Шалыто А.А. Интеграция механизма обмена сообщениями в Switch-технологию. http://is.ifmo.ru
18. Шалыто А., Туккель Н., Шамгунов Н. Ханойские башни и автоматы //Программирование. 2002. №8.
19. Шалыто А., Туккель Н., Шамгунов Н. Задача о ходе коня //Мир ПК. 2003. №1.
20. Шалыто А.А., Туккель Н.И. Преобразование итеративных алгоритмов в автоматные //Программирование. 2002. №5.
21. Корнеев Г.А., Казаков М.А., Шалыто А.А. Построение логики работы визуализаторов алгоритмов на основе автоматного подхода //Труды X Всероссийской научно-методической конференции "Телематика-2003". Т.2. СПб.: СПбГИТМО (ТУ), 2003. http://tm.ifmo.ru