

Статья опубликована в "Материалах XI Всероссийской конференции по проблемам науки и высшей школы "Фундаментальные исследования и инновации в технических университетах". СПбГУ. 2007, с. 202–205.

Шальто А.А.

ПАРАДИГМА АВТОМАТНОГО ПРОГРАММИРОВАНИЯ

*Санкт-Петербургский государственный университет информационных технологий, механики и оптики, г. Санкт-Петербург,
shalyto@mail.ifmo.ru*

Очень многие системы, которые для внешнего наблюдателя ведут себя достаточно осмысленно, представляют собой “автоматизированные объекты”. К этому классу объектов относятся люди, животные, автомобили, лифты и многое, многое другое.

Автоматизированный объект представляет собой совокупность объекта управления и системы управления, охваченных одной или несколькими обратными связями.

Идея построения автоматизированных объектов рассматривается в любом курсе теории автоматического управления или менеджмента применительно к объектам различных типов. Удивительно, но это практически не коснулось программирования.

Настоящая работа призвана устранить этот пробел. Основная идея излагаемого подхода, названного **автоматное программирование**, состоит в том, что программы предлагается создавать так же, как производится автоматизация технологических (и не только) процессов.

При этом на основе анализа предметной области выделяются источники входных воздействий и автоматизированные объекты, каждый из которых содержит систему управления (систему взаимодействующих конечных автоматов) и объект управления. Этот объект реализует выходные воздействия и формирует значения второй разновидности входных воздействий, которые от него передаются по обратным связям к системе управления.

Объект управления может быть реальным или реализованным программно. В первом случае его логика изменена быть не может, а во втором – она, при необходимости, практически вся может быть вынесена в автоматы.

Парадигма автоматного программирования состоит в представлении программ как систем автоматизированных объектов.

Универсальность предлагаемого подхода определяется тем, что он основан на расширении машины Тьюринга, которая позволяет реализовать произвольный алгоритм.

С точки зрения теории управления машина Тьюринга представляет собой автономную систему автоматизированных объектов, в которой конечный автомат, воздействующий на объекты управления (ячейки ленты), получает по обратной связи входные воздействия от управляемых объектов.

Сложность программирования на машине Тьюринга определяется низким уровнем абстракции, так как в ней используются чрезвычайно простые объекты управления, которые могут выполнять только простейшие действия (операции) по записи и стиранию отдельных символов.

В рамках предлагаемого подхода объекты управления могут быть сколь угодно сложными, а система управления может содержать не один автомат, а систему взаимосвязанных автоматов. Это обеспечивает возможность применения автоматного подхода на практике.

При написании автоматных программ предлагается разделять состояния на два класса: управляющие и вычислительные. При этом с помощью небольшого числа управляющих состояний можно управлять сколь угодно большим числом вычислительных состояний.

Основным понятием в рамках предлагаемого подхода является понятие "состояние" (управляющее). Справедливо соотношение: "Управляющие состояния + входные воздействия = конечный (детерминированный) автомат без выхода". Справедливо также: "Автомат без выхода + выходные воздействия = автомат".

Автоматы могут быть абстрактными (входные и выходные воздействия формируются последовательно) и структурными (входные и выходные воздействия формируются "параллельно"). В автоматном программировании, в отличие, например, от программирования компиляторов, применяются структурные автоматы.

Время в автоматах в явном виде не используется. Элементы задержки рассматриваются как объекты управления. При этом задержки запускаются и сбрасываются из автоматов, а информация об истечении времени поступает в автоматы в виде входных воздействий.

Стиль программирования, основанный на явном выделении управляющих состояний и применении автоматов для описания поведения программ, как отмечено выше, назван "автоматное программирование", а соответствующий стиль проектирования программ — "автоматное проектирование программ".

Автоматное программирование используется в настоящее время при логическом управлении (входные и выходные воздействия только

двоичные переменные), программировании с явным выделением состояний, объектно-ориентированном программировании с явным выделением состояний и при реализации ряда вычислительных алгоритмов и их визуализаторов.

В рамках автоматного программирования удается проводить спецификацию, реализацию, тестирование, протоколирование и документирование поведения программ в терминах автоматов. Используя этот подход, код автоматной программы может генерироваться формально и изоморфно по графам переходов.

Для поддержки автоматного программирования создано несколько инструментальных средств, наиболее известным из которых является открытое средство *UniMod*.

Это средство, в отличие от известных, позволяет реализовывать объектно-ориентированные программы с явным выделением состоянием в целом, а не только автоматы. Структура таких программ задается *UML*-диаграммами классов в форме схем связей, содержащих поставщики событий, автоматы и объекты управления. Поведение программ задается *UML*-диаграммами состояний, которые, наряду с вложенными состояниями, могут содержать и вложенные автоматы. Валидация диаграмм состояний выполняется автоматически. По диаграммам классов и состояний строится *XML*-описание программы, которое интерпретируется или компилируется совместно с вручную написанным на языке *Java* кодом для поставщиков событий и объектов управления. Изложенный подход является единственно известной автору методологией проектирования и реализации объектно-ориентированных программ с использованием *UML*-диаграмм и поддерживает такие направления современного программирования как "исполняемый *UML*" и "проектирование на базе моделей".

Автоматные программы идеально подходят для верификации методом *Model Checking*. Такие программы часто весьма удобны для реализации во многих потоках или на нескольких процессорах.

Для некоторых задач эвристическое построение автоматов невозможно или трудоемко. В этих случаях они могут строиться с помощью генетического программирования, что позволяет повысить уровень автоматизации программирования этого класса программ.

Автоматные программы могут эффективно использоваться при программной реализации, так называемых, "гибридных систем".

Из изложенного следует, что автоматный подход эффективен при построении программ со сложным поведением. Его эффективность снижается по мере упрощения логики и ее доли в общем объеме программ. Поэтому автоматный подход можно рассматривать не только как универсальный, но также и как ориентированный на учет

специфики предметной области, которой в данном случае является управление в программах.

Предлагаемый подход позволяет исправить при описании программ, говоря словами Мартина Фаулера, “ужасное состояние дел, которое мы наблюдаем сейчас в области взаимодействия программистов и экспертов в предметных областях”.

Для поддержки автоматного программирования автором создан сайт <http://is.ifmo.ru>, который, в частности, содержит более ста студенческих проектов, выполненных на основе изложенного подхода. Каждый из проектов содержит проектную документацию, что делает такие программы предметом инженерной деятельности, а не искусства.