

СИНХРОННО-АВТОМАТНОЕ ПРОГРАММИРОВАНИЕ ДЛЯ ЗАДАЧ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

С.В. Кубасов

*ГОУ ВПО «Ярославский государственный университет
им. П.Г. Демидова (ЯрГУ)», г. Ярославль*

Рецензент В.Ф. Калинин

Ключевые слова и фразы: автоматное программирование; верификация; логическое управление; синхронный подход; switch-технология.

Аннотация: Предлагается модификация технологии автоматного программирования, основанная на формальной модели и синхронном подходе. Разработаны подходы верификации синхронно-автоматных программ.

Введение

Задача логического управления возникла давно, однако до сих пор остается актуальной. Системы управления только усложняются со временем, поэтому приходится постоянно искать новые решения, отвечающие новым требованиям. Задача логического управления состоит в разработке системы, контролирующей работу некоторых объектов посредством воздействия на исполнительные устройства в зависимости от значения сигналов датчиков и желаемого алгоритма поведения.

Для решения указанного класса задач профессором А.А. Шалыто предлагается switch-технология [2], где рекомендуется активно использовать автоматы для кодирования алгоритмов логического управления. При всех своих преимуществах switch-технология не определяет четкой формальной модели, что создает значительные трудности для верификации таких программ. Очень многие задачи логического управления имеют критические требования по корректности, поэтому необходимость средств верификации таких программ не вызывает сомнений.

Настоящая статья предлагает модификацию автоматного программирования, основанную на четкой формальной модели – синхронно-автоматное программирование. Главное преимущество новой технологии – возможность автоматической верификации программ. Проверке подлежат как предопределенные свойства, так и свойства, определяемые пользователем для конкретной модели.

Кубасов С.В. – аспирант кафедры «Теоретическая информатика» Ярославского государственного университета им. П.Г. Демидова, г. Ярославль.

Синхронно-автоматное программирование

Синхронно-автоматное программирование в основном совпадает со switch-технологией. Для получения формальной модели автоматной программы пришлось уточнить некоторые аспекты оригинальной технологии и наложить дополнительные ограничения.

Новым является использование синхронного подхода. Его особенность в том, что время рассматривается как дискретная последовательность инстантов. Инстант – есть момент, в который происходит вычисление реакции системы на входные воздействия. Длительность инстанта равна нулю, все вычисления происходят мгновенно. Такой подход позволяет значительно упростить задачу верификации модели: важна не последовательность вычислений, а только результат, который в любом случае должен быть определен и однозначен.

Особенности синхронно-автоматного программирования

Синхронно-автоматная модель создавалась с учетом удобства ее использования в процессе разработки. Можно выделить такие особенности.

Модульность. Программа строится из отдельных компонентов, назовем их строительными блоками.

Инкапсуляции. Интерфейс строительного блока содержит только входные и выходные сигналы, полностью скрывая детали реализации.

Иерархическая структура модели упрощает построение сложных программ. Существующие строительные блоки могут быть повторно использованы в пределах модели.

В соответствии с гипотезой нулевой задержки *все входные и выходные сигналы присутствуют одновременно.* Отношение зависимости между сигналами приобретает условный характер.

Применение языка UML для описания модели. Разработка программы выполняется визуально.

Простота модели. Для описания программы задействовано небольшое количество понятий. Лаконичность обозначений.

Процесс разработки программы

Среда разработки. Разработка синхронно-автоматных программ производится в UML-редакторе. Используются стандартные возможности редактора, но их применение ограничивается системой правил, которая предписывает использовать только определенные типы диаграмм, определенные элементы и т.п. Контроль за соблюдением этих правил полностью остается за разработчиком.

Разработка программы ни чем, по сути, не отличается от обычного UML-проектирования. Разница только в том, что набор диаграмм интерпретируется как синхронно-автоматная модель.

Компиляция синхронно-автоматной программы. Конечной целью процесса разработки является получение системы управления. Для этого

набор UML-диаграмм необходимо подвергнуть ряду преобразований. На первом этапе выполняется общая проверка модели. Сюда включается контроль за использованием возможностей редактора, соглашения об именовании компонентов и семантические свойства модели. На втором этапе генерируется *esterel*-программа (*esterel* – язык синхронного программирования, предназначенный для разработки встроенных систем с повышенными требованиями к обеспечению безопасности). Сгенерированная *esterel*-программа в целом сохраняет структуру исходной модели. На третьем этапе выполняется проверка свойств, специфических для синхронной модели. Синхронная система должна быть детерминированной, это означает, что в любом состоянии, при любом наборе входных данных выходные сигналы системы должны определяться однозначно. Функция проверки синхронных свойств программы включена в компилятор языка *esterel*. На четвертом этапе *esterel*-программа компилируется в заданную цель. Например, это может быть программа, написанная на языке C, но существуют и другие варианты. Дополнительным этапом может быть проверка пользовательских свойств модели.

Процесс преобразования практически полностью автоматизирован. При отсутствии ошибок для получения исполняемой программы достаточно выполнить заранее подготовленную последовательность команд.

Верификация синхронно-автоматных программ

Верификация программ имеет несколько аспектов. В порядке выполнения в процессе компиляции – это проверка формата модели, синхронных и пользовательских свойств. Первые два типа проверок уже упоминались выше. Проверяемые свойства для них сформулированы заранее, поэтому сама проверка может быть выполнена полностью автоматически, достаточно запустить определенную последовательность команд.

Проверка пользовательских свойств позволяет проконтролировать выполнение тех характеристик модели, которые сложно проверить вручную. Такими являются, например, свойства, касающиеся всех возможных историй выполнения программы. Проверяемое свойство описывается отдельной программой на языке *esterel*, присоединяемой к исходной программе. К сожалению, на настоящее время разработчик синхронно-автоматных программ должен обладать знаниями языка *esterel* для описания свойств. В будущем система может быть усовершенствована для применения линейной темпоральной логики. Верификация *esterel* программ выполняется инструментом *Xeve* компании *Inria* — дополнение к набору инструментов *esterel*.

Заключение

Технология синхронно-автоматного программирования предлагает решение проблемы верификации, возникающей в автоматном программировании. Проверка пользовательских и предопределенных свойств становится возможной благодаря использованию формальной модели.

Список литературы

1. Кубасов, С.В. Синхронная модель автоматной программы / С.В. Кубасов, В.А. Соколов // Моделирование и анализ информационных систем. – 2007. – Т.14, №1. – С. 11–18.
2. Шалыто, А.А. Switch-технология. Алгоритмизация и программирование задач логического управления / А.А. Шалыто. – СПб. : Наука, 1998. – 628 с.
3. Berry, G. The foundations of Esterel / Gerard Berry // Proof, Language, and Interaction: Essays in honour of Robin Milner / edited by Gordon Plotkin, Colin Stirling and Mads Tofte. – MIT Press, 2000. – P. 425–454.
4. Bouali, A. Xeve: an Esterel Verification Environment : Version v1_3 : Rapport technique №0214 / Amar Bouali; Inria, Institut National de Recherche en Informatique et en Automatique. – France, 1997. – 23 p.

Synchronous Automatic Programming for Logical Control Problem

S.V. Kubasov

Yaroslavl State University named after P.G. Demidov, Yaroslavl

Key words and phrases: logical control; switch-technology; automata-based programming; synchronous approach; verification.

Abstract: This paper presents a modification of the automata-based programming technology built on a formal model and the synchronous approach. The technique of synchronous-automatic program verification is proposed.

© С.В. Кубасов, 2007