УДК 004.415.2

О.В. Гончаровский

Пермский национальный исследовательский
политехнический университет

РАЗРАБОТКА ПРИЛОЖЕНИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ

Рассмотрена задача создания прикладного программного обеспечения распределенных систем управления технологическими процессами. Предложена методика, основанная на представлении задачи моделью программно-управляемого автомата.

Системы управления производственными процессами, в которых элементы управления рассредоточены по всей системе и каждый узел системы оснащен хотя бы одним контроллером, называют распределенными системами управления. Для взаимодействия и мониторинга все контроллеры системы объединены в сеть. Узлы обмениваются друг с другом данными об измерениях и сигналами управления, с одной стороны, и осуществляют коммуникацию человека и машины – с другой.

Механизмы управления в распределенной системе могут быть как централизованные, так и автономные. Централизованный вариант предполагает наличие ведущего узла (мастера) – центра принятия решений. В автономном варианте решения, необходимые для каждого узла, принимаются непосредственно на этом узле, а в коммуникационной сети происходит обмен только глобальными данными. Распределение данных и функций задается непосредственно требованиями распределенного приложения.

Предметом исследования является методология (подход, метод) разработки приложения распределенной системы управления, обеспечивающая требуемую производительность. Задача поиска оптимального решения не ставится, так как даже само определение оптимального решения является достаточно сложным. Временные затраты на разработку, число единиц аппаратного обеспечения, временные характеристики и число сетевых узлов являются всего лишь несколь-

кими из множества возможных критериев оптимизации. Речь идет об усилении формализма в момент перехода от технического задания к описанию задачи (плохо описан в литературе), что должно привести к уменьшению ошибок проектирования.

Контроллеры распределенной системы управления относятся к разряду встроенных систем. Встроенные системы – системы обработки информации, установленные в конечный продукт, сегодня широко применяются в информационно-коммуникационных технологиях в качестве устройств управления.

Встроенные системы обычно являются реагирующими системами (reactive systems), такими системами, которые находятся в непрерывном взаимодействии со средой и выполняют действия, определенные средой (изменяют своё поведение в ответ на конкретные ситуации в среде).

Разработка встроенной системы начинается со знаний о прикладной области и проходит ряд этапов.

Знания прикладной области преобразуются в спецификацию – множество требований к аппаратному обеспечению, поведению, производительности, надежности, структуре, данным, отказоустойчивости, размерам, расширяемости, времени жизни, энергопотреблению, весу, дружественности, электромагнитной совместимости и т.д..

Многие аспекты спецификации встроенных систем обеспечивают модели систем, содержащие характеристики и свойства моделируемой системы, которые существенны для данной задачи.

Переход от спецификации к моделям – суть этапа разработки описания встроенных систем. Описание выполняется на том или ином формальном языке, позволяющем выполнить ее проверку на полноту, отсутствие противоречий и получить программную реализацию.

Язык описаний опирается на модель вычислений [1]. Модель вычислений представляет механизм, заданный для выполнения вычислений в виде взаимодействующих компонент. Возможные компоненты и организация в них вычислений – это процедуры, процессы, функции, конечные автоматы, а взаимодействия – коммуникационные протоколы, такие как асинхронная передача сообщений или рандеву.

Организация вычислений в компонентах:

1. Модель фон Неймана – выполнение последовательности элементарных вычислений.

2. Модель дискретных событий – события несут метки времени и выстраиваются в очередь на обработку в соответствии с ними.

3. Конечный автомат.

4. Дифференциальные уравнения – моделирование аналоговых устройств и физических систем, применяются для моделирования кибер-физических систем.

Хороший механизм для представления многих реагирующих систем дают автоматы, которые легко описывают поведение, ориентированное на состояние. Однако классические модели автоматов недостаточны для задания встроенных систем, тат как не моделируют время и не поддерживают распределенной организации поведения. Поэтому были предложены следующие усовершенствования классических автоматов:

– расширенный конечный автомат (EFSM) – автомат, пополненный переменными, которые могут быть прочитаны и записаны как часть перехода между состояниями. Введение переменных решает проблему быстрого увеличения числа состояний классического автомата для моделирования реальных объектов.

– временной автомат (TFSM) – расширенный конечный автомат, в котором часть переменных моделируют логические часы для задания временных ограничений охранных условий переходов.

– взаимодействующие конечные автоматы (CFSM) – совокупность конечных автоматов, напрямую взаимодействующих друг с другом через обмен сигналами, введены для представления распределенных систем управления.

При проектировании сложных цифровых устройств успешно применяют модель программно-управляемого автомата ПУА [2]. ПУА состоит из устройства управления (УУ) и операционного устройства (ОУ), как показано на рис. 1.

На ОУ определено множество микроопераций (МО), выполняющих преобразование входных сигналов X в выходные сигналы Y. УУ задается конечным автоматом.

Выполнение той или иной МО инициируется выходными сигналами C, УУ, зависящими от состояния УУ его входов R и S. Сигналы S характеризуют результаты выполнения операций в ОУ, а R – состояние окружающей среды.

Сигналы *S* отражают результаты выполнения МО, а *R* – состояние окружающей среды. УУ в свою очередь может быть представлено в виде ПУА и так далее.

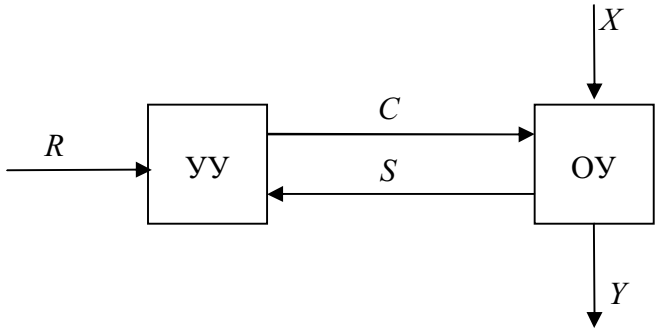

Рис. 1. Программно-управляемый автомат

Предлагается использовать ПУА для формального представления поведенческого аспекта спецификации встроенной системы и на основании этого представления выполнять разработку ее прикладного программного обеспечения. УУ, представленное как конечный автомат, на программном уровне реализуется с помощью оператора-переключателя switch (C), а ОУ – множеством программных функций по сбору и оценке измерительной информации с датчиков, а также формированию воздействий на исполнительные механизмы.

Преобразуем поведенческий аспект спецификации системы во временной автомат. Функции выходов автомата и условия переходов будут представлять ОУ.

Варианты выбора типа распределенной системы согласно Энслоу лежат между минимально распределенными системами (работа по принципу ведущий-ведомый гомогенных узлов с централизованной обработкой и накоплением данных) и полностью распределенными системами (полная совместная работа гетерогенных автономных узлов локальной обработки и накопления данных).

Рассмотрим в качестве примера простой технологический процесс, в котором задействованы три агрегата (A1, A2, A3) и рабочее место оператора (АРМ). A1 содержит актюатор a1, A2 – актюатор a2 и два датчика s1 и s2, A3 – актюаторы a3 и a4. Все актюаторы двухпозиционные.

Пусть технологический процесс задан следующим описанием**.** По команде оператора включается a1, s1 сигнализирует о достижении

измеряемой величины значения v1, выключается a1 и включается a3. Спустя время t1 выключается a3 и включается a2, s2 сигнализирует о достижении измеряемой величины значения v2, выключается a2 и включается a3. Спустя время t1 выключается a3 и включается a4. Датчик s1 сигнализирует о достижении измеряемой величины значения v3, выключается a4, и процесс останавливается до следующего пуска. Ход процесса доступен для наблюдения через АРМ.

Требуется разработать программное обеспечения для контроллеров агрегатов и АРМ как элементов полностью распределенной системы. Контроллеры и АРМ объединены локальной сетью. Контроллер агрегата А2 передает всем остальным значения датчиков s1 и s2 по мере изменения их значения.

Представим описание технологического процесса в форме временного конечного автомата. На рис. 2 приведен его граф переходов-выходов, где:

START, a1ON, a3ON1, a2ON, a3ON2, a4ON – состояния автомата;

Str, Stp – входные двоичные сигналы пуска и останова;

T – входной сигнал от таймера;

v1,v2,v3 – значения измеряемых величин;

t1 – время срабатывания таймера;

?Str=1, ?Stp=1, ?s1=v1, ?s2=v2, ?T=t1, ?s1=v3 or ?Stp=1 – условия перехода.

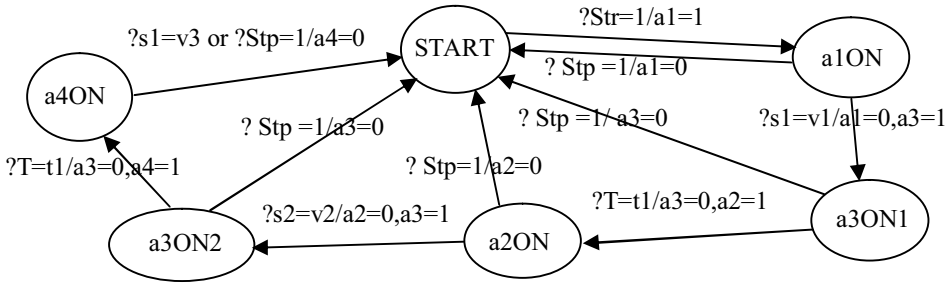

Рис. 2. Модель простого технологического процесса
в виде временного конечного автомата

Прикладное программное обеспечение каждого контроллера реализует этот автомат со своим набором функций для работы с датчиками и актюаторами. Рассмотрим работу этого распределенного приложения. В каждом контроллере все переходы должны выпол-

няться одновременно. Это должна обеспечить система синхронизации на основе корректировки часов в каждом контроллере относительно сервера времени. АРМ посылает широковещательное сообщение Str=1, которое синхронизирует часы и запускает технологический процесс. Каждый контроллер переходит в состояние a1ON, и только в контроллере агрегата A1 вызывается функция, включающая актюатор a1. Через некоторое время в контроллере агрегата A2 возникает событие s1=v1, сообщение о котором посылается широковещательно через сеть. Все контроллеры одновременно переходят в состояние a3ON1. В контроллерах запускается таймер T на время t1, после срабатывания которого все переходят в состояние a2ON и т.д.

Разработан SDK, содержащий шаблоны программ для реализации переходов временного автомата, шаблоны функций для сбора и обработки измерительной информации, а также для формирования управляющих воздействий.

Библиографический список

1. Peter Marwedel, Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems. – 2nd Edition. – Springer Science+Business Media B.V., 2011.

2. Mano M. Morris, Charles R. Kime. Logic and Computer Design-Fundamentals. – New Jersey: Prentice-Hall, 1997.

Получено 06.09.2012