

УДК 681.324

Ю.К. Апраксин, профессор, д-р техн. наук,**Т.В. Волкова, доцент, канд. техн. наук***Севастопольский национальный технический университет**ул. Университетская, 33, г. Севастополь, Украина, 99053**E-mail: kvv@sevgtu.sebastopol.ua***РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ АППАРАТНОЙ РЕАЛИЗАЦИИ ПРОТОКОЛОВ РАСПРЕДЕЛЕННЫХ ТЕХНИЧЕСКИХ СИСТЕМ**

Предлагается при проектировании протоколов распределенных технических систем (РТС) использовать автоматную модель смешанного типа [1]. В качестве основы процесса проектирования при этом используется метод структурной декомпозиции [2] первичной спецификации протокола.

Ключевые слова: *распределенные системы, протоколы, конечно-автоматная модель, структурная декомпозиция.*

Аппаратный подход к реализации систем управления распределенными объектами, в том числе протоколов их взаимодействия, является наиболее приемлемым для осуществления проектирования высокоскоростных систем передачи сообщений. Вопрос о том, какой способ реализации автоматной модели выбрать, до сих пор остается открытым. Если к системе предъявляются повышенные требования по быстродействию, то рекомендуют использовать «жесткую» логику. Выбор автоматов с программируемой логикой дает при меньшей производительности более гибкие решения. Целью настоящей работы является выработка рекомендаций по эффективному распределению функций системы управления между компонентными автоматами с жесткой и программируемой логикой.

В первом случае сложная система управления строится в виде композиции «компонентных» автоматов, совместное поведение которых эквивалентно поведению общей системы управления, заданной исходной спецификацией. При этом выделяют три способа соединения компонентных автоматов: параллельное, последовательное и с обратной связью. Именно эти способы соединения обеспечивают сохранение свойства автоматности проектируемой сложной системы управления. На рисунке 1 изображены все способы соединения компонентных автоматов и показаны преобразования, которые необходимо осуществить над алфавитами состояний (A_1 и A_2), входных воздействий (Z_1 и Z_2) и выходных сообщений (W_1 и W_2). Символом f обозначена функция формирования общего выходного алфавита при параллельном соединении автоматов, символом g – функция формирования общего входа в случае соединения с обратной связью. В случае использования автоматов с программируемой логикой необходимо разработать многоблочную многофункциональную программу выработки управляющих воздействий, инициирующих пошаговое управление общим процессом функционирования РТС (или каким-либо подуровнем этой системы).

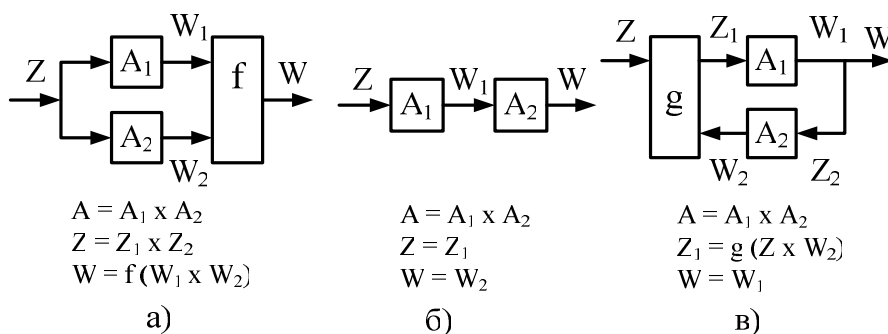


Рисунок 1 — Способы соединения компонентных автоматов, сохраняющих свойство автоматности:
а) параллельное соединение; б) последовательное соединение; в) соединение с обратной связью

В действительности оба принципа построения управляющих автоматов вполне сочетаются и могут совместно использоваться при построении единого устройства управления, состоящего из компонентных автоматов. При этом модель смешанного типа можно рассматривать как основу перспективного метода устранения основных недостатков классических управляющих устройств автоматного типа, позволяющего проектируемой системе обладать как функциональной гибкостью, так и высоким быстродействием. Под функциональной гибкостью в нашем случае понимается возможность относительно легкой модернизации. Следует отметить, что использование автоматной модели смешанного типа (жесткая и программируемая логика) наиболее простым путем обеспечивает

расширение функциональности устройства управления. Для достижения этой цели процессы, которые в дальнейшем (при эксплуатации) могут быть подвергнуты модификации для расширения их функциональности, реализуются на автомате с программируемой логикой. Автомат с жесткой логикой используется для управления общей схемой функционирования устройства управления, которая в абсолютном большинстве случаев не подвергается изменениям. Такое распределение функций по схемам управления с жесткой и программируемой логикой позволяет кроме всего прочего повысить технологичность разработки протоколов РТС.

В основу предлагаемого подхода к построению систем управления положен метод декомпозиции исходной спецификации процесса функционирования протокола РТС. Процедура декомпозиции позволяет определить допустимые варианты реализации протокола в виде компонентных автоматов. Исходным объектом для задачи декомпозиции служит ориентированный взвешенный граф $G(A, W)$, где $A = \{a_1, a_2, \dots, a_n\}$ — множество вершин и $W \subseteq \{A \times A\}$ — множество дуг. Пространством решений служит множество разбиений графа G на непересекающиеся подграфы $G_i(A_i, V_i)$, причем для каждого подграфа и для исходного графа должны выполняться ограничения (1):

$$\begin{cases} A_i \cap A_j = \emptyset & \text{для } \forall i \neq j; \\ \bigcup_{i=1}^k A_i = A; \\ \sum_{i=1}^k |A_i| = n, \end{cases} \quad (1)$$

где k — количество подграфов.

Каждый подграф G_i , входящий в разбиение, для которого $|A_i| \geq 2$, должен в структурном плане иметь:

- только одну начальную вершину (b_i);
- не менее одной конечной вершины ($e_{i,j}$), $j = 1, 2, \dots$;
- не более $n_i - 2$ внутренних вершин ($v_{i,j}$), $j = 0, 1, \dots$;
- общее количество вершин в G_i ограничено с двух сторон: $n_{\min} \leq n_i \leq n_{\max}$, причем $n_{\min} > 1$ и $n_{\max} < n$.

Под начальной вершиной подразумевается вершина подграфа G_i , входные дуги для которой исходят из вершин подграфов G_j , $j \neq i$, $j \in \{1, 2, \dots, k\}$. Конечная вершина подграфа G_i характеризуется тем, что все выходящие из нее дуги направлены только к начальным вершинам других подграфов разбиения. Внутренние вершины подграфа G_i связаны дугами только с вершинами этого подграфа. Связи между вершинами в графе G после проведения процедуры декомпозиции должны удовлетворять условиям (2):

$$\begin{cases} e_i, e_j \rightarrow b_i \rightarrow v_i, e_i; \\ b_i, v_i \rightarrow v_i \rightarrow v_i, e_i; \\ b_i, v_i \rightarrow e_i \rightarrow b_i, b_j. \end{cases} \quad (2)$$

Стрелки в (2) означают направление связи, а индекс — принадлежность к подграфу. Если условия (2) соблюдаются, то полученные после декомпозиции подграфы будем называть конструктивно-правильными разбиениями.

На рисунке 2 изображен граф автомата после проведения процедуры декомпозиции, в котором выделено три подграфа G_1 , G_2 , G_3 . Каждый подграф обозначен только своими начальными и конечными вершинами. Вершины a_0, a_4, a_5 принадлежат исходному графу G , для которого каждый подграф рассматривается как состояние автомата с жесткой логикой (a_1, a_2, a_3). Внутренние вершины подграфа и отметки дуг на рисунке не обозначены.

Следует отметить, что сформулированная задача является NP-полной, т.е. не найдено эффективных алгоритмов, позволяющих находить для нее точные решения. Предлагаемый алгоритм основан на процедуре последовательного приближения к квазиоптимальному решению. Из полученного на определенном шаге множества решений выбирается одно и делается заключение о его приближении к оптимальному. Процедура формирования решений включает процедуру построения их конструктивно

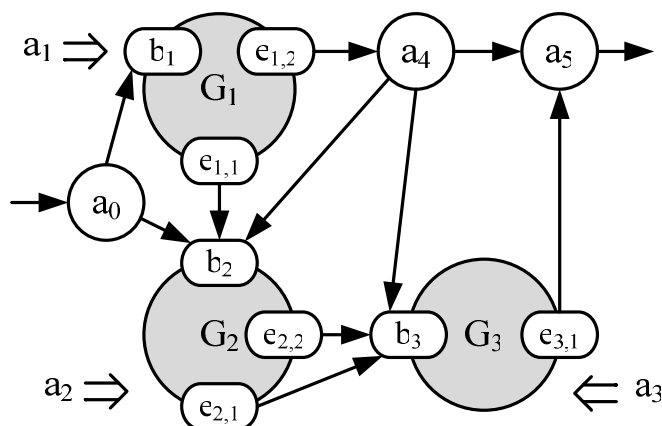


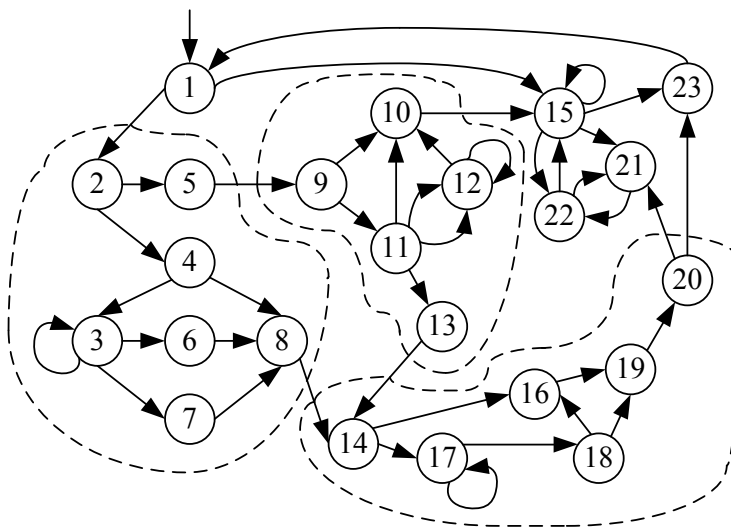
Рисунок 2 – Граф автомата с конструктивно-правильными разбиениями

правильных разбиений (условие 2) и процедуру выделения множества разбиений графа G на непересекающиеся подграфы (условие 1). Построение всех возможных конструктивно правильных разбиений обеспечивает подход [2], при котором каждая вершина a_i ($i = 1, 2, \dots, n$) графа G , кроме начальной, рассматривается как элемент нулевого разбиения. Переход на разбиения более высокого ранга (с большим количеством вершин) позволяет выделить разбиения ранга n_{min} и дает возможность строить подграфы G_i большей размерности (до n_{max}) с той же начальной вершиной b_i . Процедура выделения множества разбиений графа $G = \{G_1, G_2, \dots, G_k\}$ на непересекающиеся подграфы опирается на заданное для каждой пары графов G_i и G_j ($i \neq j$) бинарное отношение несовместимости на множестве G (3):

$$R_{ij} = \begin{cases} 0, & \text{если } (G_i, G_j) \notin R, \\ 1, & \text{если } (G_i, G_j) \in R. \end{cases} \quad (3)$$

Здесь $(G_i, G_j) \in R$ означает, что подграфы G_i и G_j несовместимы ($A_i \cap A_j \neq \emptyset$ для $\forall i \neq j$, где $i, j \in 1, 2, \dots, k$), т.е. не могут быть включены в одно множество большей мощности. Для решения поставленной задачи можно использовать алгоритм граничного перебора максимальных совместимых подграфов.

На рисунке 3 представлена первичная спецификация протокола в виде направленного графа с 23 вершинами. Нагрузки на ребрах с целью большей наглядности опущены. Граничные значения мощности множества вершин в подграфе равны: $n_{min} = 4$, $n_{max} = 7$.

Рисунок 3 – Граф G : исходная спецификация протокола

В результате выполнения процедуры выделения конструктивно-правильных разбиений определены 7 подграфов:

$$G_1 = \{b_1 = 2, E_1 = \{5, 8\}, V_1 = \{3, 4, 6, 7\}\};$$

$$G_2 = \{b_2 = 2, E_2 = \{4, 9\}, V_2 = \{5\}\};$$

$$G_3 = \{b_3 = 4, E_3 = \{3, 8\}, V_3 = \{6, 7\}\};$$

$$G_4 = \{b_4 = 5, E_4 = \{10, 13\}, V_4 = \{9, 11\}\};$$

$$G_5 = \{b_5 = 9, E_5 = \{10, 13\}, V_5 = \{11, 12\}\};$$

$$G_6 = \{b_6 = 14, E_6 = \{19\}, V_6 = \{16, 17, 18\}\};$$

$$G_7 = \{b_7 = 14, E_7 = \{20\}, V_7 = \{16, 17, 18, 19\}\}.$$

Для построения разбиения G на непересекающиеся подграфы формируем двоичную матрицу отношений несовместимости графов конструктивно-правильного разбиения (таблица 1). Матрица треугольная, так как рассматриваемое отношение симметрично.

Таблица 2 иллюстрирует динамику формирования максимально совместимых подмножеств конструктивно-правильных разбиений G .

Таблица 1 — Матрица несовместимости конструктивно-правильных разбиений

$G_i \backslash G_j$	G_1	G_2	G_3	G_4	G_5	G_6	G_7
G_1	0	0	0	0	1	1	1
G_2		0	0	0	0	1	1
G_3			0	1	1	1	1
G_4				0	0	1	1
G_5					0	1	1
G_6						0	1
G_7							0

Таблица 2 — Максимально-совместимые подмножества конструктивно-правильных разбиений G

G_1	G_2	G_3	G_4	G_5	G_6	G_7	Совмещения
1				1	1		H_1
1				1		1	H_2
1					1		$H_3 \subseteq H_1$
1						1	$H_4 \subseteq H_2$
	1				1		H_5
	1					1	H_6
		1	1		1		H_7
		1	1			1	H_8
		1		1	1		H_9
		1		1		1	H_{10}
		1			1		$H_{11} \subseteq H_9$
		1				1	$H_{12} \subseteq H_{10}$
			1		1		$H_{13} \subseteq H_7$
			1			1	$H_{14} \subseteq H_8$
				1	1		$H_{15} \subseteq H_9$
				1		1	$H_{16} \subseteq H_{10}$
					1		$H_{17} \subseteq H_1$
						1	$H_{18} \subseteq H_2$

Результатом работы второго этапа общего алгоритма нахождения подмножеств G , удовлетворяющих условиям (1) и (2), являются подмножества $H_1, H_2, H_5, H_6, H_7, H_8, H_9, H_{10}$. Выбор подмножества осуществляется на основе критерия сложности реализации разрабатываемого протокола в базисе жесткой логики и программируемой логики. Будем считать, что сложность автомата с жесткой логикой пропорциональна числу его внутренних состояний, а эта же характеристика управляющего автомата с программируемой логикой сопоставима со сложностью микропрограммного обеспечения и, в первую очередь, определяется емкостью памяти микропрограмм. В первом приближении это количество микропрограмм, реализующих различные фазы алгоритма функционирования протокола.

В таблице 3 представлены показатели сложности (число внутренних состояний управляющего автомата с жесткой логикой (УАЖЛ), число микропрограмм управляющего автомата с программируемой логикой (УАПЛ) и общий аддитивный показатель сложности) для всех возможных решений $H_1, H_2, H_5, H_6, H_7, H_8, H_9, H_{10}$.

Таблица 3 — Показатели сложности альтернативных решений

Разбиение	УАЖЛ	УАПЛ	Общая сложность
$H_1 = \{G_1, G_5, G_6\}$	9	3	12
$H_2 = \{G_1, G_5, G_7\}$	8	3	11
$H_5 = \{G_2, G_6\}$	16	2	18
$H_6 = \{G_2, G_7\}$	15	2	17
$H_7 = \{G_3, G_4, G_6\}$	10	3	13
$H_8 = \{G_3, G_4, G_7\}$	9	3	12
$H_9 = \{G_3, G_5, G_6\}$	11	3	14
$H_{10} = \{G_3, G_5, G_7\}$	10	3	13

На рисунке 4 изображено оптимальное по общему показателю сложности решение, в основу которого положено конструктивно-правильное разбиение $H_2 = \{G_1, G_5, G_7\}$. Каждый подграф разбиения на рисунке 3 выделен штриховой линией.

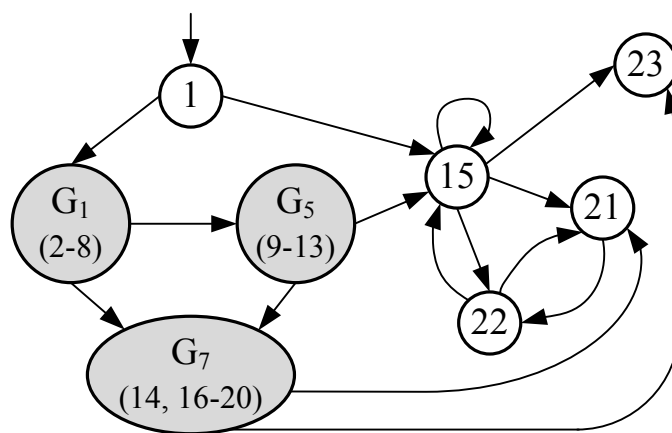


Рисунок 4 – Граф G : результат преобразования исходной спецификации протокола

Проведенный с помощью разработанной программной системы эксперимент показал эффективность предлагаемого подхода к проектированию протоколов РТС. Сложность УАПЛ и УАЖЛ соответственно $S_{ЖЛ} = f_1(|\overline{A}|, n_{\max})$ и $S_{ПЛ} = f_2(|U|, n_{\max})$ на открытом сегменте решений $[n_{\min} = 2, n_{\max} = 2, 3, \dots]$. Здесь $|\overline{A}|$ — мощность множества внутренних состояний автомата с жесткой логикой, определяющего общую схему управления и $|U|$ — общее количество переходов в выделенных в процессе решения подграфах.

Для графов с количеством дуг, незначительно отличающимся от величины $n/(n-1)/2$ в сегменте решений $n_{\min} = 2$ и $n_{\max} \in \{5, 6, 7, 8, 9\}$ при $n \leq 40$, в процессе эксперимента получено максимальное количество несовместимых конструктивно-правильных подграфов ($k = 3$). Величина $S_{ПЛ}$ на этом сегменте с увеличением $n_{\max}$ уменьшается с 20 до 8 переходов, в то время как величина $S_{ЖЛ}$

увеличивается с 6 до 30 состояний. При $10 \leq n_{max} < 20$ соответствующие характеристики сохраняют свое значение: $S_{пл} = 8 \pm 1$, $S_{жл} = 30 \pm 2$. Эксперимент, проведенный с графом, отражающим диаграмму состояний сетевого протокола TCP (Transport Control Protocol), выделил три конструктивно-правильные разбиения (фазы протокола «ожидание приема и установления соединения», «активное закрытие соединения отправителя» и «пассивное закрытие соединения получателя»), которые рекомендуется реализовать в программируемой логике с целью возможной модернизации этих фаз.

Дальнейшие исследования предполагается направить на уточнение методов оценки характеристик качества принимаемых решений и анализ качества решений для различных структур первичных спецификаций протоколов.

Библиографический список использованной литературы

1. Апраксин Ю.К. Аппаратная реализация протоколов распределенных технических систем / Ю.К. Апраксин, Т.В. Волкова // Вестник СевГТУ: сб. науч. тр. — Севастополь, 2003. — Вып. 77. — С. 71–75.
2. Апраксин Ю.К. Декомпозиционный метод определения логической корректности сложных динамических систем / Ю.К. Апраксин // Вестник СевГТУ: сб. науч. тр. — Севастополь, 1998. — Вып. 14. — С. 85–90.
3. Баранов С.И. Синтез микропрограммных автоматов / С.И. Баранов. — Л.: Энергия, 1979. — 232 с.

Поступила в редакцию 31.01.2011 г.

Апраксін Ю.К., Волкова Т.В. Розширення можливостей апаратної реалізації протоколів розподілених технічних систем

Пропонується при проектуванні протоколів розподілених технічних систем (РТС) використовувати автоматну модель змішаного типу [1]. В основі процесу проектування при цьому використовується метод структурної декомпозиції [2] первинної специфікації протоколу.

Ключові слова: розподілені системи, протоколи, кінцево-автоматна модель, структурна декомпозиція.

Apraksin Yu. K., Volkova T.V. Broadening possibilities of hardware representation of protocols of the distributed technical systems

It is suggested to use mixed automate model for distributed technical systems (DTS) design [1]. A method of structural decomposition of protocol specification is used as a basis for the design process.

Keywords: distributed systems, protocols, automata model, structural decomposition.