УДК 681.3

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОТОКОЛОВ РАСПРЕДЕЛЕННЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Апраксин Ю.К., Качинский А.С.

Севастопольский национальный технический университет
Студгородок, г. Севастополь, Украина, 99053

Рассматриваются особенности спецификации протоколов управления информационным обменом в распределенных технических системах средствами языка таблиц событий. Исследуется возможность автоматической трансляции протокольных спецификаций на язык программирования высокого уровня.

Основой для реализации протоколов управления информационным обменом в той или иной среде функционирования (программной, аппаратной или аппаратно-программной) является используемая спецификация протокола [1]. Предложенная в [2] спецификация протоколов представляет собой иерархическую систему, в основе которой лежит алфавит описания протокольных взаимодействий: $Z = \{z_k \mid z_k =< EC_k, LA_k >, k = 1,..., n\}$. Каждая буква такого алфавита характеризует k-е «элементарное событие» и определяется «элементарным условием» $EC_k = e_{1_k} \cdot e_{2_k} \cdot ... \cdot e_{r_k}$ его наступления и «элементарным действием» LA_k, которое необходимо выполнить в случае наступления соответствующего события. На перечисленные «элементарные» условия и действия накладываются требования по их совместимости, возможности и очередности применения для описания конкретного объекта протокольной системы. Эти ограничения, отнесенные к выделенной группе «элементарных событий», оформляются в виде таблиц протокольных событий. Каждая таблица (описание того или иного протокольного события) характеризует то или иное глобальное состояние протокольной системы (установление связи, передача данных, разъединение, ожидание и т.д.). Вершиной иерархической системы описания протоколов являются регулярные выражения, определяющие логику функционирования протокольной системы. Каждый объект протокольной системы взаимодействия в общем случае характеризуется своим регулярным выражением, определяющим правило поведения этого объекта, т.е. множество всех разрешенных последовательностей протокольных событий, обеспечивающих объекту взаимодействия полное выполнение возложенных на него функций.

Объектом протокольной системы являются взаимодействующие процессы и каналы, выполняющие функции передачи данных между процессами. Элементарной системой протокольного взаимодействия является система «процесс–канал–процесс». В таблице 1 приведены возможные элементарные системы протокольного взаимодействия.

Таблица 1 – Система протокольноговзаимодействия

Тип протокола	Тип канала	Объекты системы взаимодействия
Общего вида	Дуплексный	P_i, P_j, C_{ij}, C_{ji}
	Полудуплексный	P_i, P_j, C
Сбалансированный	Дуплексный	$P_i \sim \overline{P}_j$, C_{ij}, C_{ji}
	Полудуплексный	$P_i \sim \overline{P}_j$, C
Симметричный	Дуплексный	$P_i = \overline{P}_j$, C_{ij}, C_{ji}
	Полудуплексный	$P_i = \overline{P}_j$, C

Система, функционирующаяпод управлением протоколаобщего вида и использующая дуплексный канал, представляется четырьмя операторами:

P_i и P_j – взаимодействующие каждый в соответствии со своим алгоритмом процессы;

C_{ij} и C_{ji} – два канальных оператора, причем P_i использует C_{ij} только для передачи и C_{ji} только для приема данных, а P_j – наоборот.

Протокол с полудуплекснымканалом связи в протокольной системе взаимодействия имеет один канальный оператор, который используется обоими процессами P_i и P_j как для приема, так и для передачи информации, причем в каждый конкретный момент времени передачу по каналу ведет только один из процессов.

Использование сбалансированных протоколов определяет аналогичную общей протокольную систему. Однако правила взаимодействия объектов этой системы таковы, что на каждую посылочную операцию процесса P_i процесс P_j отвечает операцией приема посланного сообщения. Процессы P_i и P_j согласованы ($P_i \sim \overline{P}_j$) по последовательностям приема-передачи конкретных сообщений ($-m_r \sim +m_r$

). Такой подход существенно упрощает правило функционирования канала.

В случае симметричного протокола можно исследовать поведение только одного взаимодействующего процесса (P_i или P_j), так как поведение взаимодействующих объектов системы единообразно ($P_i = \overline{P_j}$), с той лишь разницей, что, когда P_i посылает сообщение, P_j осуществляет его прием.

Под программной реализацией протокола понимается преобразование исходной спецификации протокола в систему взаимосвязанных алгоритмических конструкций, представленную средствами языка программирования высокого уровня (ЯПВУ). При этом под элементарными действиями понимается программная процедура, выполняемая с целью организации приема (передачи) информации или других вспомогательных операций. Условие наступления элементарного события определяется функцией, возвращающей логическое значение и используемой для организации разветвлений алгоритмических протокольных процессов. Программный аналог элементарного события $z_k = < EC_k, LA_k >$ представляет собой неполный условный оператор *if* EC_k *then* LA_k. В таблице 2 приведены соответствия между конструкциями операций языка регулярных выражений (ЯРВ) и ЯПВУ.

Таблица 2 – Соответствия алгоритмических конструкций для ЯРВ и ЯПВУ

№	ЯРВ	ЯПВУ
1	$z_i \vee z_j$	*if* EC_i *then* LA_i *else if* EC_j *then* LA_j;
2	$z_i \cdot z_j$	*if* EC_i *then* LA_i; *if* EC_j *then* LA_j;
3	$z^*_k = \lambda \vee z_k \vee z_k \cdot z_k \vee \ldots$	*while* EC_k *do* LA_k;
4	$z^+_k = z_k \vee z_k \cdot z_k \vee \ldots$	*do* LA_k *until* $\overline{EC_k}$;

В первом случае формируется вложенный условный оператор, во втором – последовательность двух условных операторов, в третьем – циклическая конструкция с предусловием, в четвертом – циклическая конструкция с постусловием.

Характерной особенностью всех рассмотренных структур является то, что каждая из них имеет только один вход и один выход. Отсюда следует, что произвольный алгоритмический процесс, характеризующий систему протокольных взаимодействий, можно представить с помощью рассмотренных структур с любой степенью детализации. Кроме того, исходные протокольные спецификации могут быть существенно упрощены, для чего достаточно обратиться к аксиоматической системе ЯРВ [3].

Преобразование исходных спецификаций объектов протокольной системы взаимодействия в алгоритмические конструкции ЯРВ

осуществляется специальной программной системой – компилятором. На него возлагаются следующие функции:

1) сканирование регулярного выражения и проверка его синтаксической корректности;
2) распознавание грамматических конструкций протокольных (регулярных) выражений и подготовка соответствующих конструкций ЯПВУ;
3) анализ таблиц событий протокольных спецификаций с целью формирования:
   а) описаний объектов программного продукта (константы, переменные, массивы, пакеты, процедуры, их типы, значения и размерности);
   б) структуры используемых протоколом информационных пакетов (заголовки и информационные поля);
   в) логических функций, определяющих наступление «элементарных событий»;
   г) процедур обработки данных, связанных с каждым конкретным «элементарным событием».

Запись данных в поля заголовков производится с помощью специальной процедуры, предоставляемой компилятором при операциях установки полей заголовков каждого пакета. При написании спецификации проектировщик пользуется именами полей. Компилятор обращается к описанию заголовка пакета и определяет положение поля, его длину и допустимое значение. Набор функций, операторов и процедур, необходимых для реализации протоколов различного уровня взаимодействия объектов распределенной технической системы, задается разработчиком программного обеспечения этой системы. Все перечисленные компоненты должны составлять ядро базы данных, организованной как для проведения процедур трансляции, так и проведения экспериментов по формированию вновь проектируемых протокольных спецификаций.

Использование ЯПВУ как промежуточного языка представления протокольных реализаций позволяет получать инвариантные к конкретной операционной системе и оптимальные по трудоемкости выполняемые реализации протоколов. Решение этих вопросов возлагается на компилятор с ЯПВУ на язык конкретной программно-аппаратной среды, в которой должны будут функционировать объекты протокольной системы взаимодействия.

Библиографический список

1. Apraksin Yu.K. Computer network protocol and interface design / Yu.K. Apraksin; Helsinki University of Technology. —НТКК–ТКО–B58, 1984. — 36 с.

2. Апраксин Ю.К. Спецификация сетевых протоколов средствами языка таблиц событий / Ю.К. Апраксин // Вестн. СевГТУ. Сер. Информатика, электроника, связь: Сб. науч. тр. — Севастополь, 2001. — Вып. 31. — С. 4–8.

3. Апраксин Ю.К. Основы теории и проектирования цифровых автоматов: Учеб. пособие для вузов / Ю.К. Апраксин. — Севастополь: Изд-во СевГТУ, 2001. — 345 с.

Поступила в редакцию 8.11.2001 г.