УДК 681.3

СПЕЦИФИКАЦИЯ СЕТЕВЫХ ПРОТОКОЛОВ СРЕДСТВАМИ ЯЗЫКА ТАБЛИЦ СОБЫТИЙ

Апраксин Ю.К.

Предлагается формальная методика спецификации протоколов распределенных систем, обладающая преимуществами словесно-описательных методов.

Рассматриваемый этап проектирования (формирование формальной спецификации протокольного автомата) предполагает реализацию двух последовательных этапов. Сначала словесное (представленное средствами естественного языка) описание проектируемого протокола интерпретируется средствами языка таблиц событий [1]. Затем по полученным таблицам событий формируется формальное описание протокола в виде системы регулярных выражений, каждое из которых специфицирует один из взаимодействующих процессов (протокольных объектов).

Полученное на первом этапе описание представляет собой набор таблиц для каждого протокольного автомата [2], представляющего тот или иной объект взаимодействия протокольной системы. Так, для двух взаимодействующих объектов PAi и PAj должны быть сформированы два множества $TC^i = \left\{ TC_1^i, TC_2^i, ..., T_{k_i}^i \right\}$ и $TC^j = \left\{ TC_1^j, TC_2^j, ..., T_{k_i}^j \right\}$, причем в совокупности таблицы каждого множества характеризуют все возможные события, связанные с функционированием соответствующего протокольного автомата, т.е. условия наступления событий и совокупности процедур, выполнение которых является обязательным в случае наступления события. Кроме того, эти таблицы должны содержать описание данных, подлежащих обработке или пересылке. Структура таблицы событий показана на рисунке 1.

$EC_i(e_1,...e_r) = \tilde{e} \wedge \tilde{e}_2 \wedge ... \wedge \tilde{e}_r$ - предикат условий, в случае истинности которого должна быть выполнена процедура $LA_i(a_{l+m+1},...,a_{l+m+k})$. Выполнение процедуры $LA_i(i = 1,...,n)$ представляет собой *i*-ю реализацию события, представленного соответствующей матрицей $TC\begin{pmatrix} заголовок \\ события \end{pmatrix}$. LA_i представляет собой либо последовательность элементарных действий из множества $\{a_1,...,a_l\}$, порядок выполнения которых указывается целыми без знака в *i*-м столбце таблицы на пересечении со строкой соответствующего действия, либо процедуру $a_{l+j}!$, реализация которой представляет новое событие и описывается таблицей события $TC(a_{l+j})$ аналогичной конструкции.

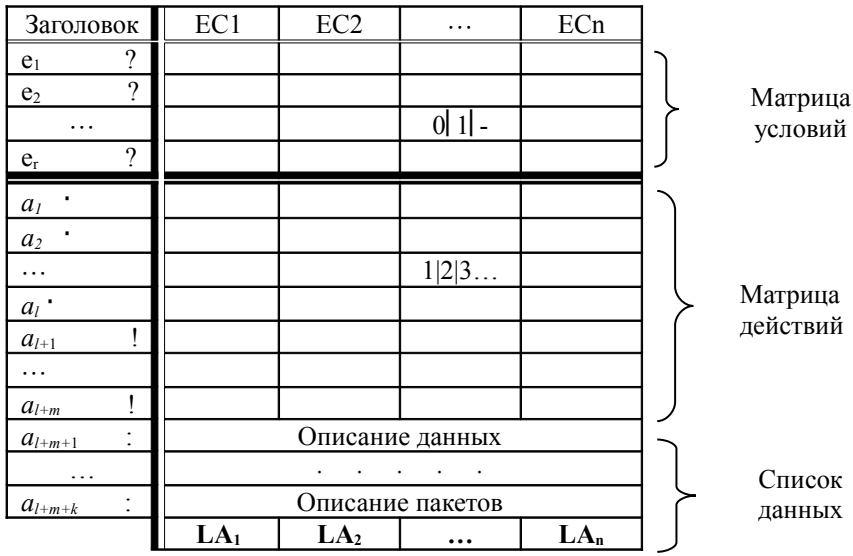

Рисунок 1 - Структура таблицы событий

Строки $a_{l+m+1},...,a_{l+m+k}$ представляют собой описание данных, подлежащих преобразованию в процессе реализации действий $a_1,...,a_l$ или пересылке.

Ограничения, налагаемые на форматы представления таблиц событий, незначительны и не превращают предлагаемый способ описания протоколов в язык математических абстракций, не всегда понятный не только неискушенному пользователю, но и не каждому проектировщику.

Предлагаемый язык спецификаций (язык таблиц событий) позволяет конструировать описание протокола в непроцедурном стиле, т.е. корректировать таблицы событий (пополнять или сокращать списки условий и списки действий) можно в произвольном порядке. Кроме того, зафиксированное в какой-либо таблице «элементарное» действие может быть переведено в разряд «сложных» действий. Для конкретизации такого действия вполне естественно должна быть разработана своя таблица событий. Другими словами, процесс спецификации протокола превращается в итеративный процесс, позволяющий на каждом шаге контролировать его результаты: полноту и безызбыточность условий, а также завершаемость протокольного процесса.

Таблица событий имеет один вход в список условий и один или более выходов (на конец алгоритма или в вызываемые таблицы событий).

На рисунке 2 изображено описание абстрактного протокольного события, представленного четырьмя таблицами событий ТС(*a*), ТС(*b*), ТС(*c*) и ТС(*d*).

a	EC_1	EC_2	EC_3	EC_4
e_1 ?	0	1	1	1
e_2 ?	-	0	0	1
e_3 ?	-	0	1	-
a_1 .	1			
a_2 .				
a_3 .	2			
a_4=*b* !		1		
a_5=*c* !			1	
	LA_1	LA_2	LA_3	LA_4

ТС(*a*)

b	EC_5	EC_6	EC_7	EC_8
e_4 ?	0	0	1	1
e_5 ?	0	1	-	-
e_6 ?	-	-	1	0
b_1 .	1			
b_2 .			1	
b_3 .			2	
b_4=d !		1		1
	LA_1	LA_2	LA_3	LA_4

ТС(*в*)

C	EC_9	EC_{10}	EC_{11}
e_7 ?	0	1	1
e_8 ?	-	0	1
c_1 .	1		
c_2 .	2		
c_3=*b* !		1	
c_4=*a* !			1
	LA_9	LA_{10}	LA_{11}

ТС(*c*)

d	EC_{12}	EC_{13}	EC_{14}
e_9 ?	0	0	1
e_{10} ?	0	1	-
d_1 .	1		
d_2 .		1	
d_3 .			1
	LA_{12}	LA_{13}	LA_{14}

ТС(*d*)

Рисунок 2 - Пример спецификации протокольного события

Первый этап формирования спецификации протокольного автомата заканчивается построением и проверкой таблиц событий. Проверка заключается в выявлении свойств

- полноты: $\vee_i EC_i = 1$, где $EC_i = \tilde{e}_1 \cdot \tilde{e}_2 \cdot \ldots \cdot \tilde{e}_n, \tilde{e}_j \in \{0,1\}$;

- безызбыточности (непротиворечивости действий): не существует EC_i такого, что $EC_i \vee EC_j = EC_j$ для любых i и $j \in \{1,\ldots,n\}$, где n – количество условий в списке условий проверяемой таблицы событий.

Второй этап построения формальной спецификации протокольного автомата заключается в построении регулярного выражения, описывающего регулярное событие, представимое форматированными таблицами собы-

тий. Алфавитом языка регулярных событий является множество $Z = \left\{ z_k \middle| z_K = \left\langle EC_k, LA_K \right\rangle \right\}$, где k =1,2,…,K и K определяет количество элементарных событий, зафиксированных во всех таблицах событий, специфицирующих протокольный автомат.

Для построения регулярного выражения, специфицирующего протокольное событие по таблицам событий, выполняется следующая последовательность шагов:

1) отмечаются все выходы таблиц событий, причем выход отмечается либо символом K (конечное состояние), если соответствующая процедура LA представляет собой последовательность элементарных действий, либо символом, соответствующим заголовку вызываемой таблицы (*a, b, c, d* – в случае предыдущего примера);

2) составляется система линейных уравнений вида

$$R_i = \bigvee_{j=1}^{K-1} R_{ji} \cdot z_{ji} \vee \delta_i, \quad i = 1,2,..., K,$$

где R_i – идентификатор множества последовательностей элементарных событий z_r, приводящих либо к событию, специфицированному ТС(i); i=1,2, …, K-1, либо к конечному состоянию K ; R_{ji} – идентификатор множества последовательностей элементарных событий z_r, приводящих к событию, специфицированному таблицей ТС(j), имеющей выход с отметкой i;
z_{ji} – идентификатор элементарного события , представляющего собой i –ю реализацию j-го события;

$$\delta_i = \begin{cases} \lambda, \textit{ если } i\textit{- идентификатор начальной таблицы} \\ \phi, \textit{- в противном случае}; \end{cases}$$

3) сформированная система линейных уравнений разрешается относительно R_K. Решение системы уравнений осуществляется методом подстановки с использованием аксиоматической системы алгебры регулярных событий и правила вывода, гласящего, что если $\alpha = \alpha\beta \vee \gamma$ и $\beta \neq \beta \vee \lambda$, то $\alpha = \gamma \cdot (\beta)^*$.

Для рассматриваемого примера:

$$\begin{cases} R_a = R_c \cdot z_{11} \vee \lambda; \\ R_b = R_a \cdot z_2 \vee R_c \cdot z_{10}; \\ R_c = R_a \cdot z_3; \\ R_d = R_b \cdot z_6 \vee R_b \cdot z_8; \\ R_K = R_a \cdot z_1 \vee R_a \cdot z_4 \vee R_b \cdot z_5 \vee R_b \cdot z_7 \vee R_c \cdot z_9 \vee R_d \cdot z_{12} \vee R_d \cdot z_{13} \vee R_d \cdot z_{14}. \end{cases}$$

Система уравнений разрешается относительно R_K:

$$R_K = (z_3 \cdot z_{11})^* \cdot (z_1 \vee z_4 \vee z_3(z_{10}(z_5 \vee z_7) \vee z_9) \vee$$
$$\vee (z_2 \vee z_3 z_{10})(z_6 \vee z_8)(z_{12} \vee z_{13} \vee z_{14})).$$

Построенные описанным методом регулярные выражения для протокольных автоматов, специфицирующих взаимодействующие процессы в распределенной системе, подвергаются анализу на предмет определения свойств полноты и завершаемости протокола.

Библиография

1 Хамби Э. Программирование таблиц решений / Э. Хамби. — М.: Мир, 1976. — 86 с.

2 Апраксин Ю.К. К вопросу автоматизации проектирования протоколов вычислительных сетей на основе конечноавтоматной модели / Ю.К. Апраксин // Автоматика и вычислительная техника. — 1987.- №2. — С.12 – 19.

Поступила в редакцию 25.09.2000 г.