

УДК 621.372

МЕТОД АВТОМАТИЧЕСКОГО СИНТЕЗА АВТОМАТНЫХ ТЕМПОРАЛЬНЫХ RVT-ГРАММАТИК²²

А.Н. Афанасьев²³, Н.Н. Войт²⁴,
С.Ю. Кириллов²⁵, В.С. Хородов²⁶

Аннотация: Предлагается метод синтеза RVT-грамматик. В основу сущности метода положено описание графических конструкций визуальных языков в виде текстовых правил.

Ключевые слова: диаграмматические модели; визуальные языки; автоматная грамматика; синтез.

METHOD OF AUTOMATIC SYNTHESIS OF AUTOMATED TEMPORAL RVT-GRAMMARS

A.N. Afanasyev, N.N. Voit,
S.Yu. Kirillov, V.S. Chorodov

Abstract: A method for the synthesis of RVT grammars is proposed. The essence of the method is based on the description of graphical constructions of visual languages in the form of text rules.

Keywords: diagrammatic models; visual languages; automaton grammar; synthesis.

1. Автоматная темпоральная RVT-грамматика

Для анализа, контроля и обработки диаграмматических моделей потоков работ на кафедре Вычислительной техники Ульяновского государственного технического университета предложен и развивается подход на основе автоматных графических грамматик. Базовым методом

²² Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-47-732152.

²³ Ульяновск, УлГТУ, e-mail: a.afanasev@ulstu.ru.

²⁴ Ульяновск, УлГТУ, e-mail: n.voit@ulstu.ru.

²⁵ Ульяновск, УлГТУ, e-mail: s.kirillov@php73.ru.

²⁶ Ульяновск, ООО РИТГ, e-mail: v.khorodov73@gmail.com.

является аппарат RV-грамматик [1]. Такие грамматики и соответствующее программное обеспечение в виде плагинов для MS Visio разработаны для большинства визуальных языков (UML, IDEF, eEPC, BPMN, SharePoint, ER, DFD и др.), используемых в практике проектирования автоматизированных систем, к которым, в частности, относятся и компьютерные системы обучения.

В ряде современных визуальных языков, в первую очередь, в BPMN используются временные параметры, накладывающие определенные ограничения на формирование последовательности событий. Для учета такой «временной» специфики предложено расширение RV-грамматики – автоматная темпоральная RVT-грамматика.

RVT-грамматика определяется шестеркой следующих компонентов:

$$G = (V, \Sigma, C, E, R, r_0),$$

где $V = \{v_e, e = \overline{1, H}\}$ – вспомогательный алфавит (алфавит операций над внутренней памятью); $\Sigma = \{a_l, l = \overline{1, L}\}$ – терминальный алфавит языка; C – конечное множество идентификаторов часов; E – множество временных выражений, определенных на C (ограничение часов и сброс часов), ограничено следующими выражениями: с начала $\{c := 0\}$ и далее $\{c \sim x\}$, причем c является переменной, а x является константой, $\sim \in \{=, <, \leq, >, \geq\}$; $R = \{r_i, i = \overline{0, I}\}$ – схема грамматики G (множество имен комплексов продукций, причем каждый комплекс r_i состоит из подмножества P_{ij} продукций $r_i = \{P_{ij}, j = \overline{1, J}\}$); $r_0 \in R$ – аксиома RT-грамматики (имя начального комплекса продукций), $r_k \in R$ – заключительный комплекс продукций. Продукция $P_{ij} \in r_i$ имеет вид $P_{ij}: a_l \frac{W_v(y_1, \dots, y_n)}{E} r_m$, где $W_v(y_1, \dots, y_n)$ – n -арное отношение, определяющее вид операции над внутренней памятью в зависимости от $v \in \{0, 1, 2\}$ (соответственно 0 – операции не производится, 1 – запись, 2 – чтение); $y_1, \dots, y_n \in V$; $r_i \in R$ – имя комплекса продукции-источника; $r_m \in R$ – имя комплекса продукции-приемника.

На рис. 1 приведен пример диаграммы на языке BPMN с темпоральной вершиной, а в табл. 1 – RVT-грамматика этого языка.

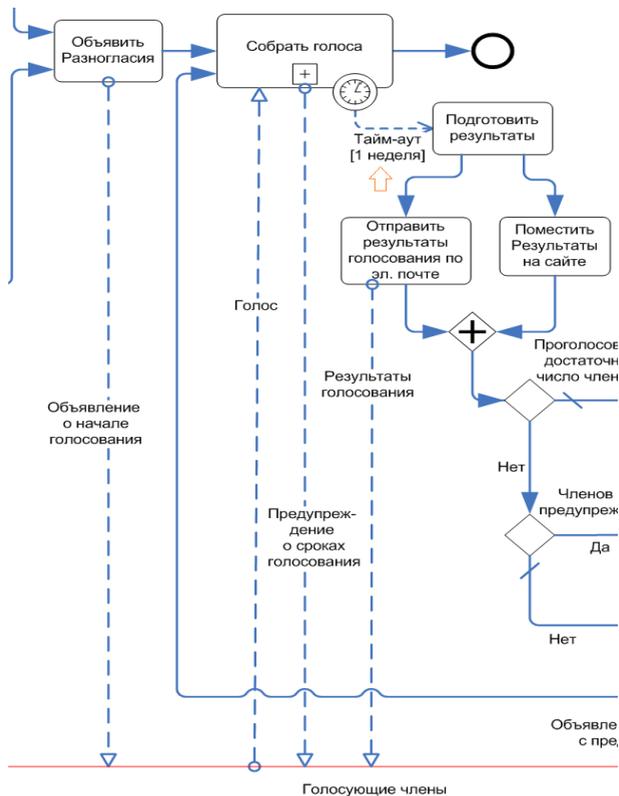


Рис. 1. Пример использования таймаута в нотации BPMN

Таблица 1. Темпоральная RVT-грамматика

N п/п	Комплекс	Квазитерм	Приемник	Операция с памятью
1	r0	A0	r1	o
2	r1	rel	r3	o
3	r2	labelEG	r3	$W_2(b^{1m}, b^{t(6)})$
4		labelPG	r3	$W_2(b^{2m}, b^{t(6)})$
5	r3	Ai	r1	o
6		Aim	r1	o
7		Ait	r1	$W_1(t_s^{t(6)})$
8		Akl	r2	$W_3(e^{1m}, e^{2m})$

Таблица 1. Продолжение

9		Ak	r4	o
10		A	r1	$W_1(t_s^{t(6)})$
11		EGc	r1	$W_1(t^{1m^{(n-1)}})/W_3(k=1)$
12		EG	r2	$W_1(I^{t(1)}, k^{t(2)})/W_3(e^{t(2)}, k \neq 1)$
13		_EG	r2	$W_1(\text{inc}(m^{t(1)}))/W_3(m^{t(1)} < k^{t(2)})$
14		_EGe	r1	$W_1(t^{1m^{(n-1)}})/W_3(m^{t(1)}=k^{t(2)}, p \neq 1)$
15		_EGme	r1	$o/W_3(m^{t(1)}=k^{t(2)}, p=1)$
16		PGf	r1	$W_1(t^{2m^{(n-1)}})/W_3(k=1)$
17		PG	r2	$W_1(I^{t(3)}, k^{t(4)})/W_3(e^{t(3)}, k \neq 1)$
18		_PG	r2	$W_1(\text{inc}(m^{t(3)}))/W_3(m^{t(3)} < k^{t(4)})$
19		_PGe	r1	$W_1(t^{2m^{(n-1)}})/W_3(m^{t(3)}=k^{t(4)}, p \neq 1)$
20		_PGje	r1	$W_1(t^{2m^{(n-1)}})/W_3(m^{t(3)}=k^{t(4)}, p=1)$
21	r4	no_label	r5	*
22	r5			

2. Метод автоматического синтеза RVT-грамматики

Алгоритмическая сущность метода основана на обработке метаописания (в виде текстовых правил) графических конструкций и их соединений визуального языка.

На рис. 2 приведены типовые конструкции диаграмм и метаописания их правил. Рамка, в которую заключены составляющие правил, обозначает границу сущности правила и иллюстрирует соответствующие входы и выходы.

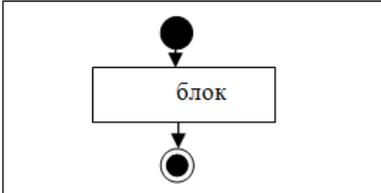
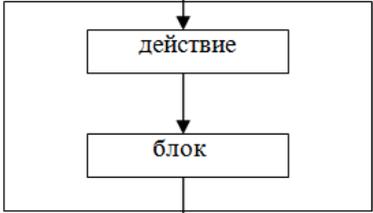
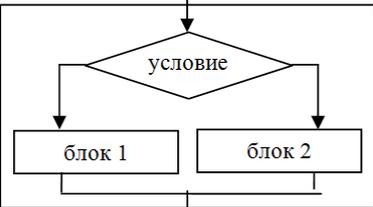
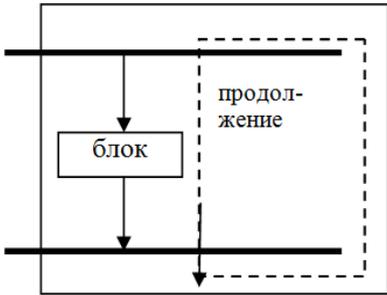
	<p>Rule Диаграмма Consist of Начало начало, Конец конец, Блоки блок Internal Relationships: блок.Вход = начало.Выход, блок.Выход = конец. Вход External Relationships:</p>
	<p>Rule Блоки Consist of Действие действие, Блоки блок Internal Relationships: блок.Вход = действие [t].Выход External Relationships: Блоки.Вход = действие.Вход, Блоки.Выход = блок.Выход</p>
	<p>Rule Блоки Consist of Блоки блок1, Блоки блок2, Условие условие , Слияние слияние Internal Relationships: условие.Выход1 = блок1 [t].Вход, условие.Выход2 = блок2 [t].Вход, слияние.Вход1=Блок1.Вход, слияние.Вход2=Блок2.Вход External Relationships: блоки.Вход = условие.Вход, блоки.Выход = слияние.Выход</p>
	<p>Rule Блоки Consist of Блоки блок, Распараллеливание распараллеливание, Слияние параллельных ветвей слияние, Продолжение продолжение Internal Relationships: распараллеливание.Выход1 = блок [t].Вход, распараллеливание=продолжение. Вход, слияние =продолжение. Выход External Relationships: блок.Вход = распараллеливание.Вход, блок.Выход = слияние.Выход</p>

Рис. 2. Типовые графические конструкции и метаописания их правил (начало)

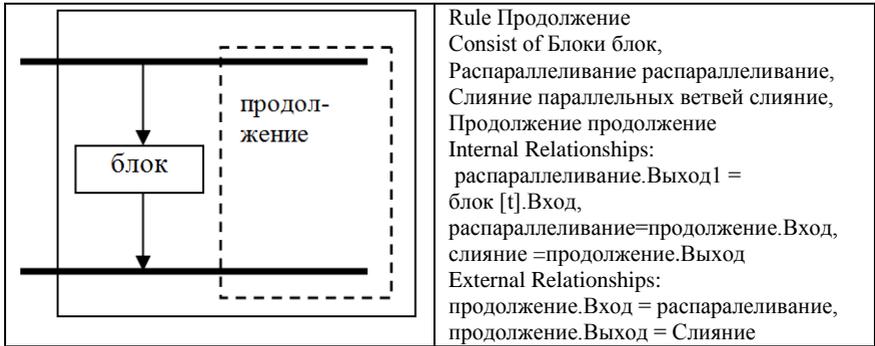


Рис. 2. Типовые графические конструкции и метаописания их правил (окончание)

Первое правило описывает диаграмму в целом. Далее правила описывают множество допустимых блоков в диаграмме. Последнее правило «Продолжение» рекурсивно ссылается на себя, благодаря чему может содержать несколько параллельных ветвей «Блоки». В объектах «Блоки» и «Действие» может быть указан параметр время t . В этом случае в правилах грамматики генерируются операции с памятью, обеспечивающие контроль этого параметра.

На основе приведенных правил генерируются терминальные и квазитерминальные символы.

Квазитермальный алфавит строится на основе части правила, начинающегося с Consist of. Построение происходит в два прохода – построение множества имен и фильтрация множества имен.

Во время первого прохода собираются все имена из правил. Для этого в каждом правиле из части Consist of строится множество используемых имен. Во время этого же прохода строится дополнительное множество имен правил. Для первого правила получим следующие множества:

Правило	Множество имен	Множество имен правил
Rule Диаграмма Consist of Начало, Конец, Блоки Internal Relationships: External Relationships: Блоки.Вход = Начало.Выход, Блоки.Выход = Конец.Вход	Начало, Конец, Блоки	Диаграмма

После окончания первого прохода необходимо отфильтровать множество имен. Целевое отфильтрованное множество равно разности Множества имен и Множества правил. Таким образом, формируется квазитерминальный алфавит блоков.

На следующем проходе формируется терминальный алфавит связей по следующему алгоритму:

- обрабатываются правила Relation;
- выделяются все типы связей для их правила;
- добавляется квазитерминальный символ в алфавит.

Синтез операций с памятью производится в следующем порядке.

У каждого правила «Блоки» есть вход и выход, следовательно, для контроля контекста можно использовать стековый механизм. В начале обработки правила в стек заносится первый элемент. В конце обработки из стека извлекается этот элемент. Если путей в правиле более одного, то возникает вопрос, когда заканчивается обработка одного правила и начинается обработка следующего.

Для контроля количества путей необходимо определить, сколько выходов имеет стартовый блок и сколько входных связей проанализировано в конечном блоке. Рассмотрим в качестве примера правило условного ветвления. Как видно из рисунка 2, из стартового элемента выходят две метки-связи. В начале обработки правила в стек записывается вершина условия, в ленту в ячейку i , соответствующую номеру условной вершины, записывается число выходов из вершины (для нашего случая 2). При достижении точки слияния в ячейку j , соответствующую номеру слияния условных ветвей, записывается значение на единицу большее предыдущего (начальное значение ноль). Когда значения в ячейках i и j совпадут, считается, что все возможные пути проанализированы, и дальнейший анализ продолжается по связи метки соответствующей выводу из конечного элемента. Для правила распараллеливания данный механизм аналогичен.

Заключение

Предложенный метод синтеза RVT-грамматики может быть использован при разработке метакомпиляторов.

Список литературы

1. Шаров О.Г., Афанасьев А.Н. Автоматная графическая грамматика // Вестник Ульяновского государственного технического университета. 2005. № 1 (29). С. 54-56.